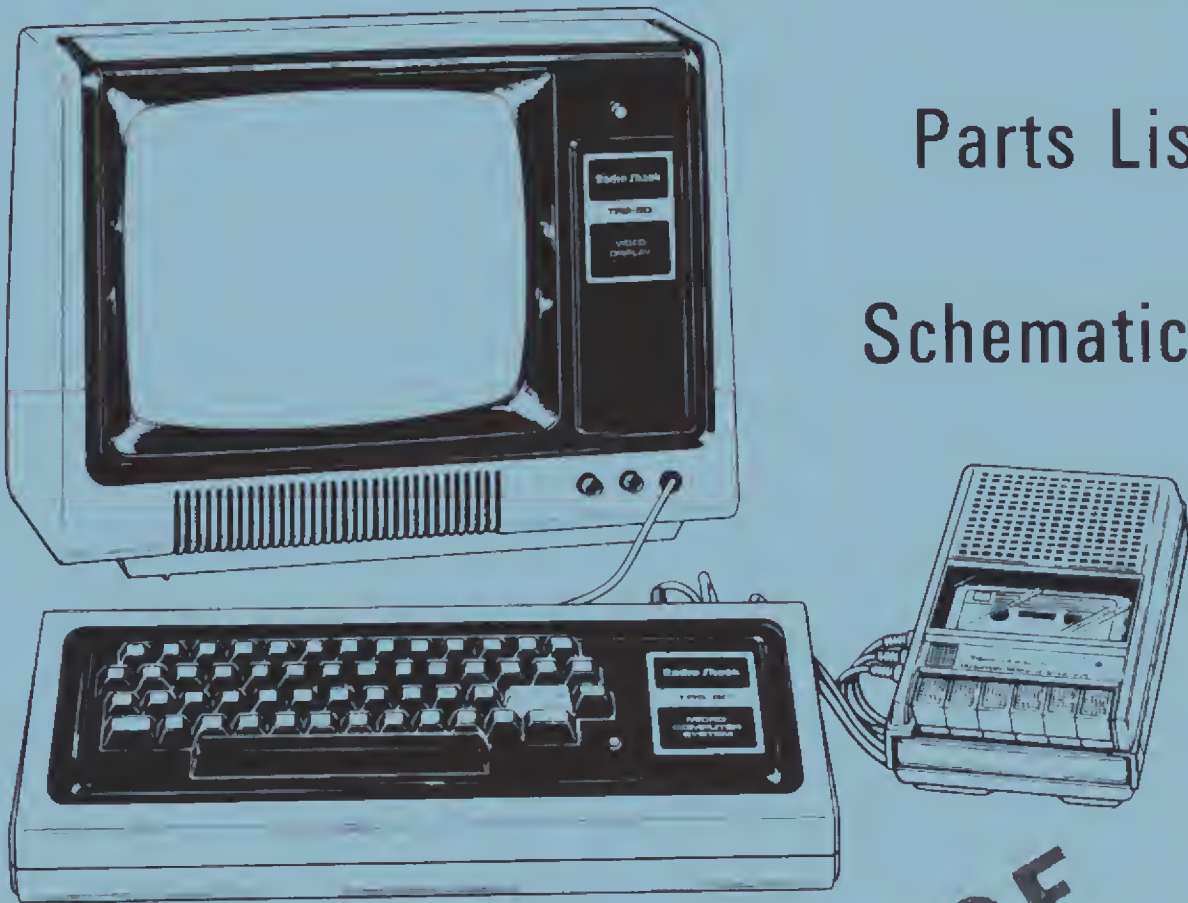


FOR
RADIO SHACK SERVICE CENTERS
TRS-80 Technical Manual

Theory

Parts List

Schematics



**PROOF
COPY**

All rights reserved. Reproduction or use, without express permission, of editorial or pictorial content, in any manner, is prohibited. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

© Copyright 1978, Radio Shack,
A Division of Tandy Corporation,
Fort Worth, Texas 76102, U.S.A.

Printed in the United States of America

TABLE OF CONTENTS

TITLE	PAGE
INTRODUCTION	2
SYSTEM BLOCK DIAGRAM	2
THE MEMORY MAP	4
THEORY OF OPERATION	5
WAIT, INT*, TEST	6
CPU Address Lines	6
CPU Data Bus	6
CPU Control Group	7
Control Group Bus	7
Address Decoder	8
System RAM	11
Video Divider Chain	13
Video RAM Addressing	16
Alphanumeric Format	16
Video Processing	17
Keyboard	23
Input and Output Port	23
Cassette Motor Control	23
Cassette Audio Output	24
Cassette Audio Input	25
INSIG*	26
System Power Supply	27
Level II ROM's	28
PARTS LIST	30
SCHEMATICS	34

INTRODUCTION

"We've done so much, with so little, for so long, that now we can do anything, with nothing, except weld the Crack of Dawn and put wheels on a Carriage Return!"

Armed with only a schematic, attacking a TRS-80 computer can be like trying to weld that crack. You may know what "CPU" stands for. You may have the knowledge of how a microprocessor system works. So you grab the schematic's book and attack that TRS-80 with soldering iron smoking and a determined gleam in your eye.

After a while, you find you have problems. You know a 4K RAM needs 12 address lines. But you only found 7. You know a computer keyboard gives you ASCII. Yet, you find the keyboard is shorting out an address line to a data line. You know how a TV typewriter scrolls characters on the screen, and you do find the video memory wired to do the job. But where is all the hardware to make the display scroll?

You know what a NAND gate symbol looks like and you also know what an OR gate symbol looks like. But some sadist has gotten all his symbols backwards. A NAND gate is shown like an OR gate, and the OR gate looks like a NAND gate. To top it all off, the power supply consists of two large rectangles with transistors and resistors sticking out of them; and the only voltages shown are the resulting outputs. Welcome to the wonderful world of computer electronics.

Now admit it. You don't really know your ASCII from a scroll when it comes to the TRS-80 computer.

The purpose of this manual is to give you a practical knowledge of system operation as it pertains to the TRS-80. This manual will show you why there are only seven address inputs to a 4K RAM. You will be shown when the microprocessor inputs data from the keyboard the CPU thinks the keyboard is a memory, of all things! You'll learn how the CRT screen is scrolled and you might even learn to appreciate the backward symbolization. As far as the power supply is concerned, you might find that it's not nearly as complex as you thought. So, grab your schematics and let's take a tour of the TRS-80 computer.

SYSTEM BLOCK DIAGRAM

The 80 integrated circuits contained in the TRS-80 can be broken down into about 10 major sections. Figure 1 shows these sections as they relate to other sections. The heart of the system is definitely the CPU (Central Processing Unit). You might consider the CPU as being a very dumb calculator circuit. It may be dumb, but it's a fast dummy. Most of the leads on the CPU are data lines and address lines. The CPU tells the address bus where the data it wants is located, and the data bus is a good place for the information to

come back to the CPU. The address lines are outputs from the CPU. They never receive data or addresses from other sections. The data lines on the other hand can give or receive data.

ROM

The ROM (Read Only Memory) could be considered the brains, if the CPU has to be the heart, of the system. The ROM tells the CPU what to do, how to do it and where to put it after it's done. Without the ROM, the CPU would just sit there and oscillate. When power is first applied to the system, the CPU has just enough smarts to output an address to the ROM that locates the CPU's first instruction. The ROM shoots back the first instruction and then the two really start communicating. In less than a second, the CPU, under ROM supervision, performs all the house-keeping necessary to get the system alive and a "READY" flashes on the screen:

If the CPU misses that first piece of ROM data, then it may go bananas. It may tell the ROM that it is ready to load a tape so the ROM tells it how to do that. The tape recorder turns on. But since the CPU is now playing games in the video memory, who cares about the tape? The CPU operates at about 2 MHz; therefore, digital screw-ups seem instantaneous.

Remember that the CPU is the work horse and the ROM is the boss. The ROM tells the CPU how to do it, when to do it, and where to put it.

RAM

The next major section in Figure 1 is the RAM (Random Access Memory). This memory is where the CPU may place data it doesn't need until later. The RAM is also the place where the programs are kept for use. If you tell the computer to count to 10,000, then the CPU stores your instructions in the RAM. If you tell the computer to do it NOW, here is what happens:

The CPU tells the ROM someone wants in. The ROM tells the CPU to go to the keyboard and find out who. The CPU finds out, tells the ROM that it's the boss. The ROM tells the CPU to find out what he wants. The CPU tells the ROM that the boss wants US to RUN. The ROM tells the CPU to go to RAM and find out what the boss wants done. The CPU says the boss wants to count to 10,000. The ROM tells the CPU how to do it. After it's done, the ROM tells the CPU to find out what to do with it. The CPU informs the ROM that the 10,000 has got to go on the display and must be saved. The ROM tells the CPU how to put it on the display and then tells it to store the 10,000 somewhere in RAM; but it had better remember where it is. The CPU tells the ROM that the job is done. The ROM tells the CPU to monitor the keyboard in case the boss wants something else.

The CPU looks to the ROM for instructions. The CPU then follows the ROM's instructions and looks to the keyboard, then the RAM. In all cases, the CPU applies address locations to the ROM, RAM, and keyboard. The data lines are then checked for input data that corresponds to these address locations. In case of an output from CPU to RAM, the CPU selects the address, puts data on the data lines, then instructs the RAM to store the data that is on the data lines.

Notice that only the CPU communicates with all other sections. If the CPU is told by ROM to store something from ROM into RAM, the CPU can't make the RAM receive ROM data direct. Instead, the CPU takes the data from ROM and then sends it to RAM. The CPU must act as intermediary between the two. The reason for this is that the CPU is the only section that can address locations and pass data to all other sections.

KEYBOARD, VIDEO RAM, VIDEO PROCESSING

The keyboard section is not necessary as far as the CPU is concerned, but it is very necessary for the operator. The keyboard is our method of making known our instructions to the CPU. The opposite is true for the video RAM. In this case, the CPU wants to tell us it needs data or it may want to show us the result of a complex calculation. So, the request for more information or the result is stuffed into the video RAM. Anything in video RAM is automatically displayed on the monitor. The video processing section handles this. Data in the video RAM is in ASCII. Converting ASCII into the alphanumeric symbols we recognize is the job of the video processor. A ROM contains all of the dot patterns. The ASCII locates the character pattern, and the video processor sends it out to the terminal.

VIDEO DIVIDER CHAIN

Composite video going to a terminal is extremely complex. Aside from the video signal, there is the horizontal and vertical sync. These signals must be very stable and be outputted in the correct sequence. The CPU is busy enough as it is, so the video divider chain handles the TV work. It generates the sync signals and addresses the video RAM in a logical order so that the video processor can handle video data efficiently. Notice the block under the video RAM labeled MUX. This is short for Multiplexer. It acts somewhat like a multipole, multiposition switch. When the video divider chain is in control, the MUX is switched so that only addresses from the divider chain are directed to the video RAMS. The CPU may need to read or write data into the video RAM. If so, the MUX is switched so that the CPU has control over video RAM's address. After the CPU is finished, the addressing task is reassigned to the divider chain.

THE MEMORY MAP

Some customers call the Factory Customer Service Center and ask "Which output port is the display?" These customers are told that the TRS-80 does not use an output port for the display. They are told that the TRS-80 is memory-mapped. In a memory-mapped system an address will define and select all other subsections.

Figure 2 shows the memory map for a Level I TRS-80. From memory locations 0000 to 0FFF, the Level I ROMs are present. The keyboard is located from address 3800 to 380F. The video display is located from 3C00 to 3FFF. The RAMs start at 4000 and, depending on how much RAM is in the system, can extend down to address 7FFF.

HEX ADDRESS	DESCRIPTION OF CONTENTS/USAGE
0000 To 0FFF	Level I ROMs
1000 To 37FF	Not used
3800 To 380F	Keyboard
3810 To 3BFF	Not used
3C00 To 3FFF	Video Display
4000 To 41FF	RAM Used by Basic Level I
4200 To 4FFF	Useable RAM starts here
5000 To 5FFF	RAM
6000 To 7FFF	RAM
8000 To FFFF	Not used

NOTE: Map not drawn to scale

Figure 2. Level I Memory Map

As stated before, upon power-on, an address location is outputted by the CPU requesting information from the ROMs. Since the ROMs are located at addresses 0000 to 0FFF, the CPU will be outputting addresses in this area. If the CPU needs some kind of keyboard data, it will output addresses 3800 through 380F and see if anything is in this "memory" location. If the CPU wants to show the programmer something on the display, all it has to do is address the video display section of the map and store data in these locations. Something to remember: the video display shows exactly what is in memory locations 3C00 through 3FFF.

Notice memory locations 4000 to 41FF. At address 4000 RAM starts. But, part of the RAMs are used by Basic I as general housekeeping memory locations. Hence, the user accessible RAM actually starts at address 4200.

THEORY OF OPERATION

System Clock

The system clock is shown on Sheet 2 of the fold-out schematics in the Schematic Section. Y1 is a 10.6445 MHz, fundamental cut, crystal. It is in a series resonant circuit consisting of two inverters. Z42, pins 1 and 2, and 3 and 4, form two inverting amplifiers. Feedback between the inverters is supplied by C43, a 47 pF capacitor. R46 and R52 force the inverters used in the oscillator to operate in their linear region. The waveform at pin 5 of Z42 will resemble a sine wave at 10.6445 MHz. The oscillator should not be measured at this point, however, due to the loading effects test equipment would have at this node. Z42, pin 6, is the output of the oscillator buffer. Clock measurements may be made at this point. The output of the buffer is applied to three main sections: the CPU timing circuit, the video divider chain, and the video processing circuit.

CPU Timing

The Z80 microprocessor needs a single phase clock source for operation. The 10 MHz signal from system clock is applied to Z56, a standard ripple counter, which is used as a divide-by-6 counter. The resulting signal at Z56, pin 8, is a little over 1.774 MHz. The signal is applied to the input of buffer Z72, pin 12. Pin 11 of Z72 is attached to pin 6 of the Z80 microprocessor. R64 pulls up pin 11 of Z72, and insures a rapidly increasing rise time for the clock signal. Notice that pin 15 of Z72 is tied to ground. Since pin 15 is the enable input to this part of Z72, pin 12 and 11 will always be active. Notice also pins 7 and 6 of Z56. These two pins enable the clear function for the counter. When one or both of these pins is low, the counter operates normally. When high, the input forces the counter into its clear or reset state. Z42, pins 9 and 8, are used to disable counter Z56 during automatic testing at the factory. R67 pulls Z42's input to V_{CC}, which causes pin 8 to stay at a logical low. During testing, pin 9 of Z42 may be pulled low, making pin 8 high, which disables and clears Z56. You might also find early Board levels (A Boards for example) where pins 6 and 7 of Z56 are tied directly to ground.

Power-Up-Clear and System Reset

As mentioned in the block diagram discussion, upon power-on the CPU accesses a known address in the ROM for instructions. The circuitry which causes the starting address output is shown just below the microprocessor clock divider. Z53 is a 2-input, quad NAND gate. (Note that Z53 is drawn like an inverted input OR gate.) When power is first applied to the system, C42 is at 0 volts. R47 is tied to V_{CC} and starts charging C42 at a known rate. While C42 is charging, and before the voltage exceeds the logical 1 level for Z53, pin 11 outputs a high. This high is inverted by Z52, pins 11 and 10, and a low is applied to pin 26 of Z40.

A low at this input forces the microprocessor to output the starting address 0000 on its 16 address lines. When C42 charges up past about 1.4 volts, Z53, pin 11, goes low, which causes Z52, pin 10, to go high. The CPU is now out of its reset state, and will start executing instructions from the ROM, starting at address 0000. Notice that the only time pin 26 of the CPU is ever low is a few milliseconds after power is applied. Once C42 charges up past the logical ONE level, pin 26 stays high until C42 is discharged when power is removed. Why is Z53, a NAND gate, drawn like an OR gate? Notice that pin 11 is high only when either of the inputs are low. The NOT circles at the input immediately tell you that this gate is looking for a signal that is low to cause an output that is high. Had the gate been drawn "correctly", then it would not have been so obvious that the output is active when high. This "functional" type of logical symbolization is used throughout the schematics.

Directly above the power-up circuit, there is a similar circuit. S2 is the reset switch located on the right side of the Board. Although there is a power-on-delay type circuit on the input of this network, it is not used as such. Notice that C57 is smaller than C42. Hence, in a power-up "race", C57 would charge up faster than C42. Assume that C57 is charged. Also assume that pin 2 of Z53 is high. This means that Z53, pin 3, will be low and Z37, pin 13, will be high. With pin 17 of the CPU held high, everybody is happy. If S2 is pressed, C57 will discharge through the switch. The resulting low is applied to pin 1 of Z53 and pin 3 goes high. Z37, pin 13, is then forced low. A low at pin 17 of the CPU forces the microprocessor to restart at address 0066. When S2 is released, R65 begins to charge C57 until a logical high is applied to pin 1 of Z53. At this time, pin 17 of the CPU goes back high and the CPU starts executing instructions from address 0066 in the ROMs.

S2 is used to get the microprocessor back on the right road when it is "lost". This switch forces the CPU toward a known address to enable it to get on the right track. An example of a lost CPU would be during a bad cassette load attempt. If a cassette is loading and suddenly there is missing information on the tape (caused by dirt or age), the recorder may never stop. S2 can then be pressed, which directs the CPU out of the cassette load routine and back into its ready mode.

The output at pin 18 of Z40 is called "Halt". In Level I BASIC, this output should never be low. It goes low only when a software halt instruction is encountered by Z40. In theory, this instruction is not included in the ROMs. But you might find pin 18 held low because Z40 thought it was told to halt. It could be due to some data malfunction, or the CPU is lost and is playing around with display data instead of ROM data. In a case like this, S2 is not effective in bringing the CPU home, because Z53 is latched up. About all you can do is shut the computer down and try again.

CPU ADDRESS LINES

Notice that Z53, pin 11 and pin 3, are also tied to Z37, pins 2 and 3. Z37, pin 1, is an output line labeled SYSRES* (System Reset Not). It is normally-high and only goes low during power up (Z53, pin 11, causes this), or when S2 is pressed (Z53 pin 3, causes this). SYSRES* is used by the expansion interface and is not used by the TRS-80 in BASIC I.

One last thing to mention about these two circuits: When you turn off power to the TRS-80 because of a lost CPU, wait at least 10 seconds before you reapply power. If you do not wait, C42 may not discharge all the way and the CPU may not go back to address 0000 during a restart. By waiting, C42 will discharge and upon power-up, the system will start at the correct ROM location.

WAIT, INT*, TEST

These three inputs to the CPU are pulled up to VCC through resistors. Since they are active low, you may not have any use for them. But you should know what they are for.

The WAIT input, pin 24 of Z40, will slow the CPU down if there are slow memories it must access. If this line goes low, the CPU will go into a wait status until it goes back high. Once high, the CPU continues with the operation. For example: Assume you have a memory system that takes 100 microseconds before addressed data can be guaranteed to be present at the output. When the memory logic sees that the CPU wants data, it will make the WAIT line low. At the end of 100 microseconds time, the logic will make the WAIT pin high, and the CPU will input the data.

The INT (Interrupt Request) is at pin 16 of Z40. This input when low, will force the CPU into an interrupt request section of the memory. It would then perform some instruction associated with the interrupt. An example of this use would be as follows: Assume there was a door on the back of the TRS-80 that should always be closed. There was a switch connected to the door such that when opened, the switch contacts are shorted. The switch would be connected to ground and to pin 16 of Z40. If the door were opened, the computer would stop what it was doing and print on the screen "Close Door." The CPU would be interrupted, and it would henpeck you until you closed that door! As you can see, pin 16 is tied to VCC through a resistor and is not used. It is, however, used with the TRS-80 Expansion Interface.

The TEST input may be quite useful in your troubleshooting. Pin 25 of Z40 is labeled $\overline{\text{BUSRQ}}$ (Bus Request). When this pin is brought low, it will force the data, the address and the control lines into the disabled or floating state. Although it is not used by the TRS-80 in normal operation, it is quite useful when someone wants to "shut down the CPU". We'll talk about this input when we discuss the Control Logic Group.

There are system outputs of the microprocessor labeled A0 through A15 that start the address bus. Since these lines must go to ROM, RAM, the keyboard and the video RAM, they must be buffered for two reasons. First, the buffers must be able to supply the address bus with proper logical levels. The microprocessor cannot supply the current necessary to drive all of the sections connected to the address bus, and buffers are needed for current gain. Secondly, it may be necessary to switch off the address bus. For example, if an Expansion Interface is connected to the bus, it may be necessary to address RAM in the main unit for a data transfer. Therefore, there must be some method to take the CPU off the data bus. The buffers are tri-state devices. This means they will either act as buffers or as opened switches.

Z38, Z39 and part of Z22 and Z55 are the address line buffers. Notice that in Z38 and Z39 there are two sections of buffers. The first section contains four buffers and the second section contains only two buffers. Each section is controlled by a single pin. The first is controlled by pin 1 and the second by pin 15. When these control pins are at a logical low, the buffers are enabled and will operate normally. When the control pin is at a logical high, the buffers are disabled, and will show a high impedance from input to output. The signal that controls the address buffers is labeled "ENABLE*" and is sourced at Z52, pin 4. Pin 3 is the input for control line inverter, Z52, and is tied to the TEST* line. Notice that R58 keeps this line pulled high. Hence, the address buffers' control line will always be at a logical low; and therefore, operating as buffers. If TEST* is shorted to ground, the address buffers will be disabled. This feature could be very useful in troubleshooting.

CPU DATA BUS

The data bus is buffered like the address bus, except for one area. Notice that there are only eight data lines at the CPU, labeled D0 through D7. But there are 16 buffers. Remember that the CPU must receive data as well as send data. The address lines are strictly CPU outputs, while the data lines are inputs and outputs. Therefore, there must be two sets of buffers for the data line. One set handles CPU output data while the other set takes care of the CPU input data.

The output data buffers consist of all of Z75 and one section of Z76. The input buffers consist of one section of Z55 and the last section of Z76. Notice that the input and output buffers are connected "head to toe". This could cause problems if both were on at the same time! The control inputs to the output buffers are all connected together on the line labeled DBOUT*, and are in turn tied to Z53, pin 6. Likewise, the input buffers' controls are tied together on the line labeled DBIN*, and are connected to Z53, pin 8. DBOUT*, is tied to pins 9 and 10 of Z53, the gate which generates DBIN*.

As you can see, Z53, pin 6, is the major source of input or output data control. If pin 6 is high, \overline{DBOUT}^* is high and \overline{DBIN}^* is low. Therefore, the input buffers are enabled and the output buffers are disabled. If Z53, pin 6, is low, \overline{DBOUT}^* is low and \overline{DBIN}^* is high. In this case, the output buffers are enabled and the input buffers are off.

Pin 4 of Z53 is tied to \overline{TEST}^* . If \overline{TEST}^* is grounded, not only will we disable the address buffers, but we will also cause pin 6 of Z53 to go high. Hence, the data output buffers will be off, robbing the CPU's control over the data lines. Since \overline{DBIN}^* is now held low, the input data buffers would be active. But, this would not cause any problem since the address bus from the CPU has been disabled.

When \overline{TEST}^* is left alone, it is held high. If pin 21 of the CPU (the Memory Read output) is high, Z53, pin 6, will be low. The low causes \overline{DBOUT}^* to be low and \overline{DBIN}^* to be high. Therefore, the CPU is outputting data; and the buffers are switched accordingly. When pin 21 of Z40 goes low, Z53, pin 6, will be high. We now have almost the same condition as if \overline{TEST}^* went low. \overline{DBOUT}^* is high and \overline{DBIN}^* is low but the address buffers are still enabled. The data buffers are now ready for the CPU to accept data.

CPU CONTROL GROUP

OK, we now know how the CPU accesses the address bus. We know the data bus is used to gather data into the CPU or pass data out of the CPU. What we do not know at this point is how the CPU stores data in a memory or how it tells the ROM or RAM that it is ready to receive data. The CPU control group performs this task. These signals are: RD, WR, OUT, and IN.

RD (Read)

RD is Read control. This signal, when activated, will tell other sections that the CPU is ready to accept data. RD is generated at Z23, pin 6. Pin 5 is connected to pin 21 of Z40, the \overline{RD} (Read) output. Pin 4 of Z23 is tied to pin 19, \overline{MREQ} (Memory Request), of the CPU. Therefore, when pins 19 and 21 of Z40 go low at the same time, an RD output is generated. Notice the backward symbol for an OR gate. It's drawn like an AND gate. When we get \overline{MREQ} and \overline{RD} , then and only then will we get RD. We're looking for two lows on the input for a low output.

WR (Write)

WR is Write control. This signal, when activated, will tell other sections that the CPU is ready to write data into one of the memory locations. WR is generated at Z23, pin 11. Pin 12 of Z23 is connected to \overline{MREQ} . Pin 13 of Z23 is tied to \overline{WR} (Memory Write), which is pin 22 of Z40. When we get a low at the \overline{MREQ} output and a low at the \overline{WR} output, then and only then will we get a low at WR.

OUT (Output)

OUT is Output control. This signal, when activated, will enable circuitry to perform the cassette save functions. It would also be used to control data movement from the TRS-80 to the Expansion Interface. OUT is generated at Z23, pin 3. Pin 1 of Z23 is tied to the \overline{WR} output on the CPU. Pin 2 of Z23 is tied to \overline{IORQ} (Input/Output Request) which is pin 20 of the CPU. When we get a low at \overline{WR} and a low at \overline{IORQ} , then and only then will we get a low at OUT.

IN (Input)

IN is Input control. This signal, when activated will enable circuitry to perform the cassette load function. It would also be used to control data movement from the Expansion Interface to the TRS-80. IN is generated at Z23, pin 8. Pin 10 of Z23 is connected to \overline{IORQ} . Pin 9 of Z23 is tied to \overline{RD} . When we get a low at \overline{RD} and a low at \overline{IORQ} , then and only then will we get a low at IN.

CONTROL GROUP BUS

The Control Group must be buffered for use by the different sections. Also, the bus may need to be switched off at some time. Therefore, part of Z22 is used to buffer the Control Group. Tri-State control at pin 1 is tied to the address bus control, and \overline{ENABLE}^* will affect the status of the address and the control group bus in the same manner.

ADDRESS DECODER

As shown in Figure 2, the TRS-80 is memory mapped. Therefore, the address 01AC (in HEX) is in the ROM part of the map. Address 380A is in the keyboard area and 3CAA accesses the video display RAMs. Since the data and address buses are connected in parallel to all the sections, there must be some method to determine which section is being accessed. A decoding network monitors the higher order address bits and selects which "memory" the CPU wants to use. The address decoder is so important to the operation of the system that it has been redrawn in Figure 3. Keep your schematic handy since there are signals shown in Figure 3 that need to be sourced or traced.

The address decoder uses six bits. A10 through A15 are needed plus RD* and RAS* (RAM Address Select). A15 is the most significant bit of the address bus. Let's combine the six high order bits and add a couple more, so that we have two hex digits:

A15 A14 A13 A12 A11 A10 A9 A8

A12 through A15 form the most significant hex character. A8 through A11 form the next most significant hex character. A8 and A9 are the two bits we had to add to complete that last hex character. Now let's break down part of the memory map into hex and binary.

	A15	A14	A13	A12	A11	A10	<u>A9</u>	<u>A8</u>	
From: Hex 0000	0	0	0	0	0	0	0	0	Level I ROMs
To: Hex 0FFF	0	0	0	0	1	1	1	1	
From: Hex 3800	0	0	1	1	1	0	0	0	Keyboard
To: Hex 380F	0	0	1	1	1	0	0	0	
From: Hex 3C00	0	0	1	1	1	1	0	0	Display RAMS
To: Hex 3FFF	0	0	1	1	1	1	1	1	
From: Hex 4000	0	1	0	0	0	0	0	0	4K RAM
To: Hex 4FFF	0	1	0	0	1	1	1	1	
From: Hex 4000	0	1	0	0	0	0	0	0	16K RAM
To: Hex 7FFF	0	1	1	1	1	1	1	1	

Notice in the breakdown that we could use the two most significant digits of the hex code in the decoding scheme and handle the selection of all the memories. In the binary columns, you can see that instead of using two hex digits, which is eight binary lines, we can ignore two bits and use only six binary lines. A dotted line separates the two unused bits from the six that we'll use.

Looking at Figure 3, you'll see that bits A12, A13 and A14 are connected to Z21, a dual, 2-input to 4-line decoder/demultiplexer. The C1 and C2 inputs are connected in such a way as to make Z21 into a 3-input to 8-line decoder. The G1 and G2 inputs to Z21 are chip enables. As shown, when these inputs are at a logical 0, Z21 is active. When high, Z21 is disabled and none of its eight outputs are low. The G enables are controlled by OR gate Z73, pins 4, 5, and 6. Pin 4 is tied to A15, the most significant bit of the address bus.

Notice in the memory map breakdown that A15 is always low when addressing the various memories. Z73, pin 5, is tied to RAS* (Row Address Select). Go back to the big schematic, Sheet 1, and find MREQ at pin 19 of the CPU. As stated earlier, MREQ only goes low when the CPU needs or wants to output memory data. Follow pin 19 down to Z72, pin 4. This buffer sources RAS* and it is the same signal as MREQ. Back to Figure 3. When A15 and RAS* are low at the same time, a low will be outputted by Z73, pin 6. This low will enable Z21. When Z21 turns on, one of its outputs will go low, depending on the status of A12, A13 and A14. For example, if these three inputs are at logical zero, pin 9 will go low. If all three inputs are high, pin 4 will go low. You might consider A12 through A14 as supplying an octal address to Z21. Since there are eight states in an octal code, then there could be one of eight lines selected (Output 0 through Output 7).

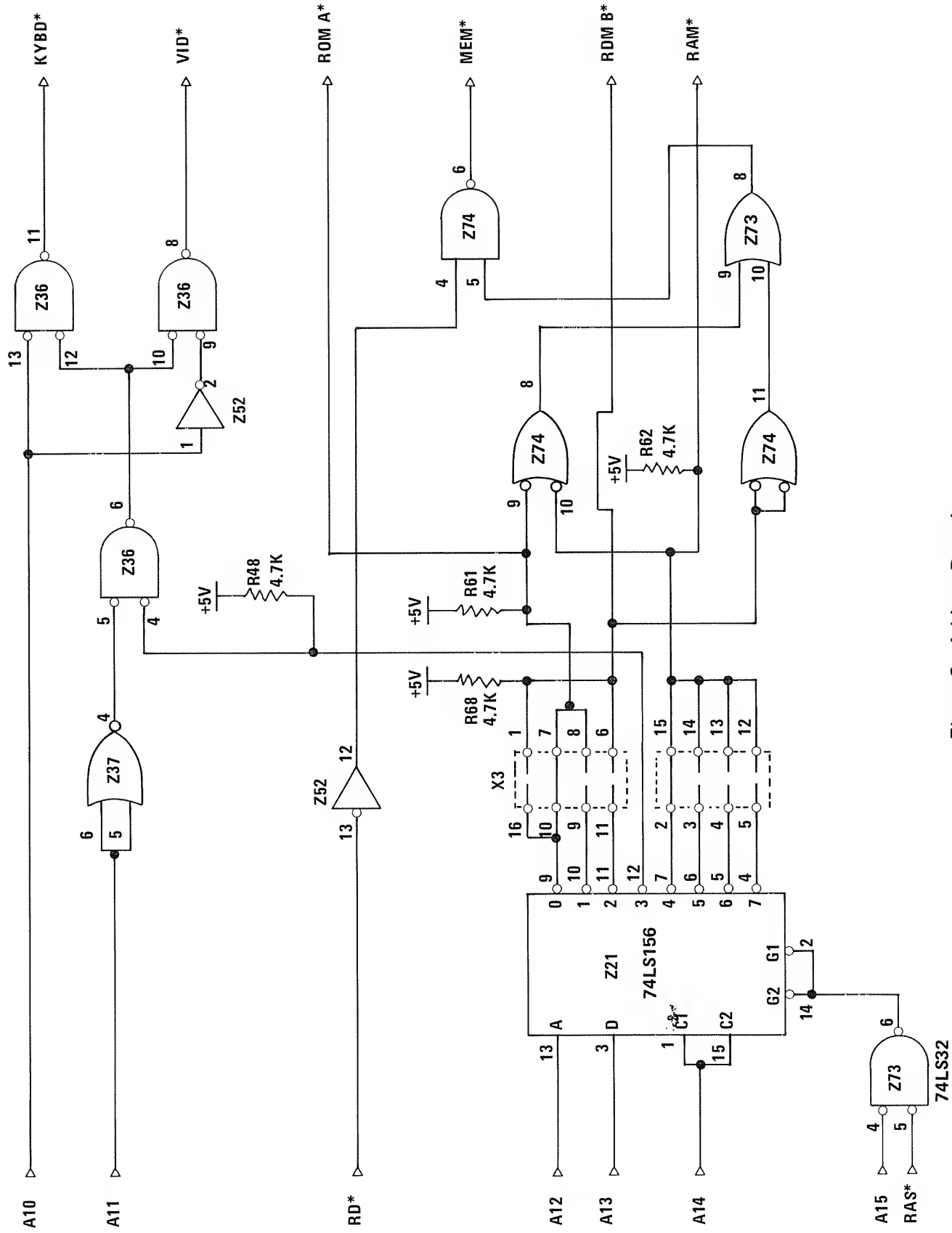


Figure 3. Address Decoder

KEYBOARD DECODING

We can sum up Z21's function quite simply: It decodes the most significant digit of the hex address. Using Z21 and the last two bits, A11 and A10, we can define any one of the four "memories" available to the CPU in BASIC I.

ADDRESS DECODER PROGRAMMING

Attached to the outputs of Z21 is X3. X3 is called a "DIP shunt" and it is installed in the PCB position Z3. A DIP shunt is like a shorting bar array, except the bars may be broken. By breaking some bars and leaving others intact, the address decoder is programmed to reflect the amount of RAM or ROM the CPU has available for use. In Figure 3, X3 is shown with six broken shorting bars. We will use this configuration in our discussions.

ROM DECODING

When the CPU needs instructions on how to perform a certain task, it must access ROM. ROM Decoding is performed as follows: The CPU needs a memory, so RAS* will go low. The address for ROM starts with hex 0, so A15, A14, A13, and A12 go low. Z21 becomes active due to the low at A15 and RAS*. Pin 9 of Z21 goes low. Follow pin 9 through the shorted bar at X3 pins 10 and 7, past the pull-up resistor R61 and out to ROMA*. If you find ROMA* on the large schematic, you'll see it goes to ROM A, (Z33, pin 20.) This pin is the CS (Chip Select) and it is active low (as the inverting circle on pin 20 shows). Z33 turns on, which means that its output becomes active (Note: the ROM's outputs are tri-stateable like the buffers. When CS goes low, the ROM outputs will switch from a high impedance or off-state to an on-state. When on, the outputs will go low or high depending on the data in the ROM at the address specified.) We got the address applied to ROM A and we got ROMA* to go low, so ROM A is turned on. But now we need to insure a data path is opened so that we can pass data from ROM to CPU. Notice in Figure 3, ROMA* is also attached to pin 9 of NAND gate Z74. A low on pin 9 will cause a resulting high at pin 8. Z74, pin 8, is tied to Z73, pin 9. Z73, pin 8 passes a high to Z74, pin 5. Z74, pin 4, is tied to RD*, part of the CPU control group. Since the CPU is trying to read data from ROMs, RD* will be low. Pin 4 of Z74 will then be high, because RD* is inverted by Z52, pins 13 and 12. OK, we know pins 4 and 5 of Z74 are high, so that makes pin 6 low. This low is MEM*. If you find MEM* on the big schematic, you will notice it controls the ROM/RAM buffers. The outputs of the buffers are tied to the data bus. We now get ROM data onto the data bus. Has it got a way to get to the CPU? Yes, it does. Remember that RD is low because the CPU is in a Memory Read cycle. Since this is so, DBIN* is low and DBOUT* is high. The low at DBIN* enables the CPU's input data buffers and ROM data is available for the CPU.

The keyboard is located from address 3800 to 380F. The keyboard is memory, so RAS* will be low. A15 is low because we are generating address codes under 8000. Looking at our binary location for the keyboard, we find A14 low, A13 and A12 high. With this input combination, Z21 will be active and pin 12 will be low (Output 3). Pin 12 is tied to Z36, pin 4. According to the breakdown, A11 is high. Z37, pin 4, outputs a low to Z36, pin 5. The "incorrectly drawn" OR gate tells us we need both inputs low for a low output. We got it, so pin 6 of Z36 is also low. Pin 6 of Z36 is tied to pins 12 and 10 of Z36. Checking on the status of A10, we find it listed as being low during a keyboard address output. Since Z36, pins 12 and 13 are low, we'll get a low at pin 11. KYBD* is generated at this pin.

Finding KYBD* on the big schematic, you'll see it goes to the enable inputs of the data buffers for the keyboard. The lower order address lines are tied to one end of the keyboard matrix, while the other end of the matrix is tied to the data bus, through the buffers. If a key is pressed, an address line will be "shorted" to a data line. Assume for now that this scheme works. We'll analyze the keyboard later. The DBOUT*, DBIN* signals are switched the same way as if we had a ROM select. Therefore, keyboard data will get to the CPU's data bus for processing.

VIDEO DISPLAY RAM SELECT

In the binary breakdown for the memory map, you will notice that the binary out for the video RAM address is almost the same as the keyboard except for bit A10. Z21, pin 12, will output a low to Z36, pin 4. Since A11 is still high, Z37, pin 4, will supply a low to pin 5 of Z36. Therefore, pin 6 of Z36 is low, just as if a keyboard was selected. Since A10 is now high, Z36, pin 11, is high and KYBD* is not active. But Z36, pin 10 is low and so is pin 9 due to the effects of inverter Z52, pins 1 and 2. Hence, Z36, pin 8, goes low and we have caused VID*, the Video RAM select, to become active. Assume for now that VID* does select the video RAMS. We'll discuss what it does and how it does it later.

4K RAM DECODER

As shown on the memory map, the address which selects RAM extends from hex 4000 to 4FFF for 4K. The binary breakdown lists the states of A15 as a 0, of course. A14 is high and A13 and A12 are low. We are still accessing memory, so RAS* is low. Hence, Z21 will be active and output 4 will be low (pin 7). DIP shunt X3 passes this low through pins 2 and 15, and it is applied to Z74, pin 10. It also is outputted by the decoder section as RAM*. RAM* will select the CS pin on all of the RAMs, after it passes through DIP shunt X72. (It's shown on sheet 2 of the large schematic.)

The selection of the data bus for RAMs is handled the same way during a ROM-Read operation. MEM* will go low because RD* went low. But during a CPU data dump from CPU to RAM, MEM* does not select the data bus buffers for the RAM. Instead of RD* being active, WR* is low. We don't need the ROM/RAM buffers because the RAM data inputs are on the output side of the buffers. Only during a ROM/RAM read operation do we need MEM*.

Notice on X3 that we can program the system for 8K of RAM by leaving the shorting bar intact at pins 3 and 14, and at the 2 and 15 position. Not only would a 4000 address cause RAM*, but a 5000 address would also enable RAM*. If we had 12K of RAM, we would leave pins 4 and 13 shorted. For 16K, we short all pins we have mentioned; plus pins 5 and 12. RAM* would now be active from addresses 4000 to 7FFF.

As you can see from the RAM discussion, we'll be shorting certain outputs of Z21 together. In most applications, using TTL, shorting output nodes is bad design practice. But there are some TTL devices that are called "open collector" types. These types of gates do not have an active pull-up on the output. Instead, the output transistors have "open collectors". It is the responsibility of external circuitry to pull them up. The "Open collectors" outputs may be tied together for a "wire OR" function.

Since Z21 is an open collector decoder, the output may be safely tied together. Notice resistors R48, R61, R62, and R68. These are the pull-up resistors for Z21. Something to remember about open collector outputs: You cannot tell if one of these outputs is working unless there is a pull-up resistor tied to that output. For example, if you placed an oscilloscope probe on pin 10 of Z21 as shown in Figure 3, you would not be able to tell if pin 10 goes low. If the system is working right, it shouldn't. But if it isn't working right and pin 10 is going low, how are you going to prove it? Pull it up with a resistor to +5 volts and see; that's the only way you can be sure.

SYSTEM RAM

According to the block diagram, system RAM is tied in parallel with the data bus and address bus just like ROM and the keyboard. The data input and output for RAM is straightforward enough; MEM* controls the buffers. But the addressing scheme appears all screwed up. How can the CPU address a minimum of 4K of RAM using only seven address inputs? The answer to that very good question is — multiplexing. The address from the CPU is multiplexed into the RAM in two 7-bit parts. The RAM's internal logic takes the two parts and brings them together to form one address scheme with 14 bits. One part of the addressing is called RAS* (RAM Address Select); the other part is CAS* (Column Address Select). Another signal, MUX (Multiplexer), controls the switching function. All three of these signals are generated near the CPU on Sheet 1 of the schematic section.

MUX, CAS*, RAS*

On Sheet 1, find pins 21 and 22 of the CPU. Follow the lines tied to these two pins down to NAND gate Z74. If we get a low at WR (Memory Write) or a low at RD (Memory Read), Z74, pin 3, will output a high (called MREQ, Memory Request). MREQ is tied to the clear inputs of Z69 and part of Z70. These devices are D type flip-flops where the MUX* and CAS* signals are generated. Figure 4 shows a waveform chart for this circuit. Line A shows the master clock input to the flip-flops. Line B shows MREQ and Line C depicts the WR output from the CPU. Assume that the CPU wants to write data into RAM. As shown on line B, MREQ will go low. A short time later, WR will go low. Line D shows Z74, pin 3, going high at the same time as WR went low. The flip-flops now have a logical high applied to the clear inputs. The flip-flops are free to operate, controlled by the clock waveform. On the next rising edge of the clock, Z69, pin 5, will output the logic level that was present at pin 2 the instant that pin 3 went high. Since pin 2 was high when pin 3 went high, pin 5 will go high. This high is shown on line E. Z69, pin 12, is now high; so on the next rising edge of the clock, pin 9 will go high. This is shown on line F. Z70 is ready to toggle. On the next rising edge of the clock, Z70, pin 6, will go low (Q went high, so Q-bar must go low). This is shown on line H of Figure 4. All three flip-flops have changed states since WR went low. The flip-flops will stay in this state so long as WR stays low. When WR does go high, the flip-flops will have a low applied to their clear inputs; and they will reset back to the clear condition.

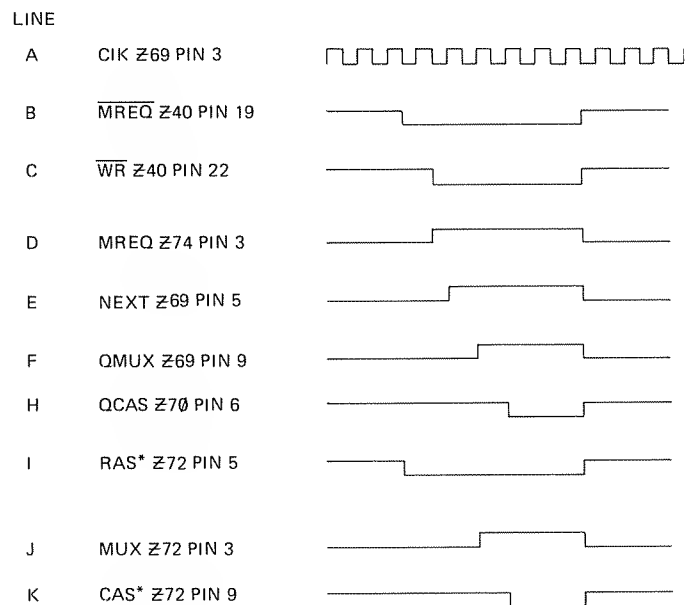


Figure 4. Waveform Chart

Line I is the \overline{RAS}^* output. As you can see, it is a direct function of \overline{MREQ} from the CPU. Z72, pins 4 and 5, is \overline{RAS}^* 's buffer. Line J, MUX is sourced at Z69, pin 9, through buffer Z72, pins 2 and 3. Line K, \overline{CAS}^* , is buffered by Z72, pins 10 and 9, and is a function of QCAS, at Z70, pin 6.

Notice the following sequence of events: \overline{RAS}^* goes low first. MUX then changes states. \overline{CAS}^* then changes states one clock cycle later. First, we get RAM address select, then MUX, then we get Column Select. Hence, the first part of the address we give the RAMs will be the row address. We'll then flip the switch (multiplexer) and follow with the column address.

RAM ADDRESSING

In about the middle of sheet 1, on the left side of the RAM array, multiplexers Z35 and Z51 are shown. On the left side of Z35, we find the area where four address lines are coming in. One brace of four is labeled "0" and the other is labeled "1". Z51 is configured the same way except there are only three lines per brace. The 0 tells us that when the select pin is low, the multiplexer will be outputting data associated with these input lines. The "1" tells us the opposite is true. When the select pin on the multiplexer goes high, it will be outputting data associated with the "1" input address lines. The select input for both multiplexers is pin 1. Z35, therefore, operates somewhat like a 4-pole, double throw switch, where the select input (pin 1) is doing the switching. Z51 is used only as a 3-pole, double throw switch — one input/output is not used. The enable input to the multiplexer is pin 15. Since pin 15 is grounded on both IC's, the "switches" are always enabled.

READING FROM RAM

Assume the CPU needs RAM data. Let's follow the addressing and data paths the RAM will use. We'll use a 4K RAM example.

The CPU outputs a \overline{MREQ} and a \overline{RD} . The address decoder outputs \overline{MEM}^* and \overline{RAM}^* . \overline{MEM}^* activates the RAM/ROM data buffers and \overline{RAM}^* enables the chip select (\overline{CS}) for the RAMs. At the same time, the multiplexer will load the address into the RAMs. \overline{RAS}^* goes low. The MUX signal is low at this time, so A0 through A5 on the RAM receive the low order address. Notice that \overline{RAS}^* is buffered by Z68, pins 14 and 13, and is applied to pin 4 of all the RAMs. The negative going signal at pin 4 will load the lower order address in the row section of each RAM. A short time later, MUX changes state; it goes high. The multiplexer, Z35 and Z51, now switch and the high order addresses are applied to the RAMs. \overline{CAS}^* will now go low. \overline{CAS}^* is applied to buffer Z67, pin 14. Pin 13 of Z67 passes \overline{CAS}^* to pin 15 of all eight RAMs. On the negative transition of \overline{CAS}^* , the high order addresses (A6 through A11) will be loaded in the

column section of each RAM. The RAMs now have the entire address from the CPU. The RAM will now output data through the buffers and to the CPU.

WRITING TO RAM

The difference between a write operation and a read operation is exactly two signals. Address decoding and address multiplexing work the same way. During a data write, however, the CPU sends data to the RAMs. Hence, the ROM/RAM buffers are not needed; and \overline{MEM}^* will not go low. Instead of the CPU issuing a \overline{RD} command, it supplies a \overline{WR} instruction. \overline{WR}^* is tied to all eight RAMs on pin 3. When this pin is low, data will be stored in RAM at the specified address. When this pin is high, the RAMs are in read cycle.

REFRESHING THE RAMS

The TRS-80 uses dynamic type RAM. A dynamic RAM differs slightly from a static RAM in data retention. A static RAM will retain data stored in it so long as power is applied to it. A dynamic RAM must be periodically addressed to ensure that it retains the data loaded into it. The periodic addressing is called refreshing. You might compare a dynamic RAM with an air-filled tire with a slow leak. Every once in a while, the tire must be shot a little air so it won't go flat. If we did not service that tire, it would finally become unusable. The same is true about dynamic RAM. If the system does not access the RAMs every so often, they will "forget" data.

The dynamic RAM in the TRS-80 uses an " \overline{RAS} Only" type of refresh. In other words, when \overline{RAS}^* goes low, the RAMs in the system will "refresh themselves" even though the RAM may not be in use at the time. As stated before, \overline{RAS}^* is generated by the CPU at pin 19 (\overline{MREQ}). Whenever \overline{MREQ} goes low, \overline{RAS}^* goes low; and the RAMs will load the lower order address into the row section. The CPU may be looking at system ROM when \overline{MREQ} goes low, but RAM will still receive \overline{RAS}^* and hence be "refreshed."

Normally, you would not be too concerned about this aspect of the RAMs. But you need to be aware of the differences between a static RAM and a dynamic RAM. Remember: Dynamic RAM must be periodically addressed to enable it to retain data. In the TRS-80, the RAM should be refreshed once every two milliseconds.

RAM PROGRAMMING

You may have noticed X71 during the discussion of the RAM. X71 is a DIP shunt. It is used to program the size of memory in a system. Find pin 13 on the RAM. Following pin 13 down, you will see it is tied to two pins of DIP shunt X71. Pin 13 of the RAM is the CE (Chip Enable) or the A6 address input. In a 4K system, pins 4 and 13 of X71

are shunted. RAM* is on pin 4 so RAM* is used to select RAMs. But in a 16K system, 4 and 13 are opened and pins 3 and 14 of Z71 are shorted. Instead of RAM*, we'll get address line A6 or A12 (depending on multiplexer status) going to pin 13 of the RAMs. There are other parts of X71 shown on the left side of the multiplexer, Z35 and Z51. Before troubleshooting a system, you will need to know the size of RAM the system uses. If X71 is "programmed" wrong, you may find yourself with RAM problems.

VIDEO DIVIDER CHAIN

The video divider chain supplies the video RAMs with addresses in a logical order for video processing. This chain also supplies the horizontal and vertical sync timing pulses so that the video processor can build the composite waveform for the display. Video RAM addresses, horizontal and vertical sync, and video processing timing are all direct functions of the master clock. Also included in the divider chain is the hardware necessary to generate 32 character line lengths. Although BASIC I can not access the 32 character format, BASIC II can.

DIVIDER CHAIN – INPUT CONDITIONING

If the TRS-80 did not have to change character line formats, the divider chain could have been tied directly into the master clock. But, the TRS-80 does have two formats for character lengths. In the most familiar format, the display has 16 character lines, each consisting of 64 characters. This means there are 1024 character locations in video RAM the divider chain must access. In the other format, the characters appears twice as large. The display will show 16 character lines of 32 characters. The divider chain must access only 512 video RAM locations. Switching from one format to the other is the job of the input conditioning logic.

On sheet 2 of the schematic section, the master oscillator circuit is surrounded by a D flip-flop (Z70), a divide-by-12 counter (Z58) and a multiplexer (Z43). The D flip-flop is wired to perform a divide-by-two function. The multiplexer is wired such that we can route the master clock frequency, or the clock frequency divided by 2, from the flip-flop to the divide-by-12 counter. Since there are two character length formats, there must logically be two reference frequencies; one is half again as slow as the other. The master oscillator supplies the divide-by-12 counter with a reference frequency in a 64 character format. The D flip-flop supplies the counter with the reference frequency in a 32 character format.

The multiplexer is doing the selecting, so what is controlling it? Pin 1 of Z43 is a signal called MODESEL (Mode Select). When low, MODESEL forces Z43 to be switched into its 32 character position. When high, MODESEL forces Z43 to be switched into its 64 character position. Let's look at the 64 character mode first.

Since MODESEL is high, pin 3 is "shorted" to pin 4 of Z43. Pins 6 and 10 are "shorted" to pins 7 and 9. (Remember: a multiplexer is an electronic equivalent of a multi-pole, double throw switch.) Figure 5 is a waveform chart for this circuit. At line A, the master clock is shown at the output of its buffer, Z42. Line B shows the action of D flip-flop during its divide-by-2 function. The buffered clock is applied to pin 3 of Z43. Since the multiplexer is switched into its "1" state, pins 3 and 4 are the same signal and counter Z58 receives the 10 MHz clock frequency at pin 14. Notice that flip-flop output Z70, pin 9, is tied to pin 2 of Z43. It is not performing any function at this time since the multiplexer is not switched into its "0" state.

The output of Z58 is shown at lines C, D, E, and F in Figure 4. The arrows in this figure points out the place where Z58's outputs are all zero. Notice that lines C through F do not count up to 11, then go back to zero using straight binary. Z58 starts fine: 0 1 2 3 4 5 On the next clock, it goes from binary 5 to binary 8. From 8, it counts normally to binary 13; then on the next cycle, it goes back to binary zero.

Notice pins 6 and 7 of Z58. These inputs are used to clear the counter to zero. If you find CTR on sheet 1, you will see it comes from inverter Z42, pin 8, which controls the CPU CLK divider. Normally, CTR is held low. Only during automatic testing at the factory is CTR allowed to go high and clear Z58. You might find "A" and "D" Level Boards with Z58, pins 6 and 7, simply tied to ground.

Z58, pin 12, is labeled DOT 1. Z58, pin 9, is labeled DOT 2. DOTs 1 and 2 are "nanded" by Z24; and the resulting output is shown in Figure 5 at line G. This signal is called "LATCH" and is used in video processing.

Z43, pins 6 and 10, are tied together and are connected to Z58, pin 8. The resulting output is Z43, pins 7 and 9, will therefore be the same signal. Pin 9, labeled "CHAIN" is the divider chain's main source. Pin 7 of Z43 is labeled "C1" and is tied to pin 10 of Z64, one of the video RAM multiplexers. C1 will be used to address the video RAM's least significant bit.

In the 32 character format, Z43, pin 1, will be low. Therefore, pins 2, 5 and 11 will be "shorted" to pins 4, 7 and 9 respectively. (The electronic switch was flipped.) Now we have the frequency source from Z70, pin 9, tied to counter Z58. Pin 7 of Z43 is held low all the time; and pin 9 of Z58 is now used as the source labeled "CHAIN". In Figure 5, lines H through K show Z58's outputs. Remember: We are using line B in the Figure as the input to Z58 instead of line A. Notice that Z58 is now being used as a divide-by-6 counter. The output at pin 9 is now "CHAIN" instead of pin 8. Has the CHAIN frequency changed? No. In 64 character mode, we had the master clock, divided by 12, as the chain frequency. That is 10.6445 MHz divided by 12 = 887.041 KHz. In 32 character mode, we had 1/2 master clock divided by 6, as the chain. 10.6445 MHz

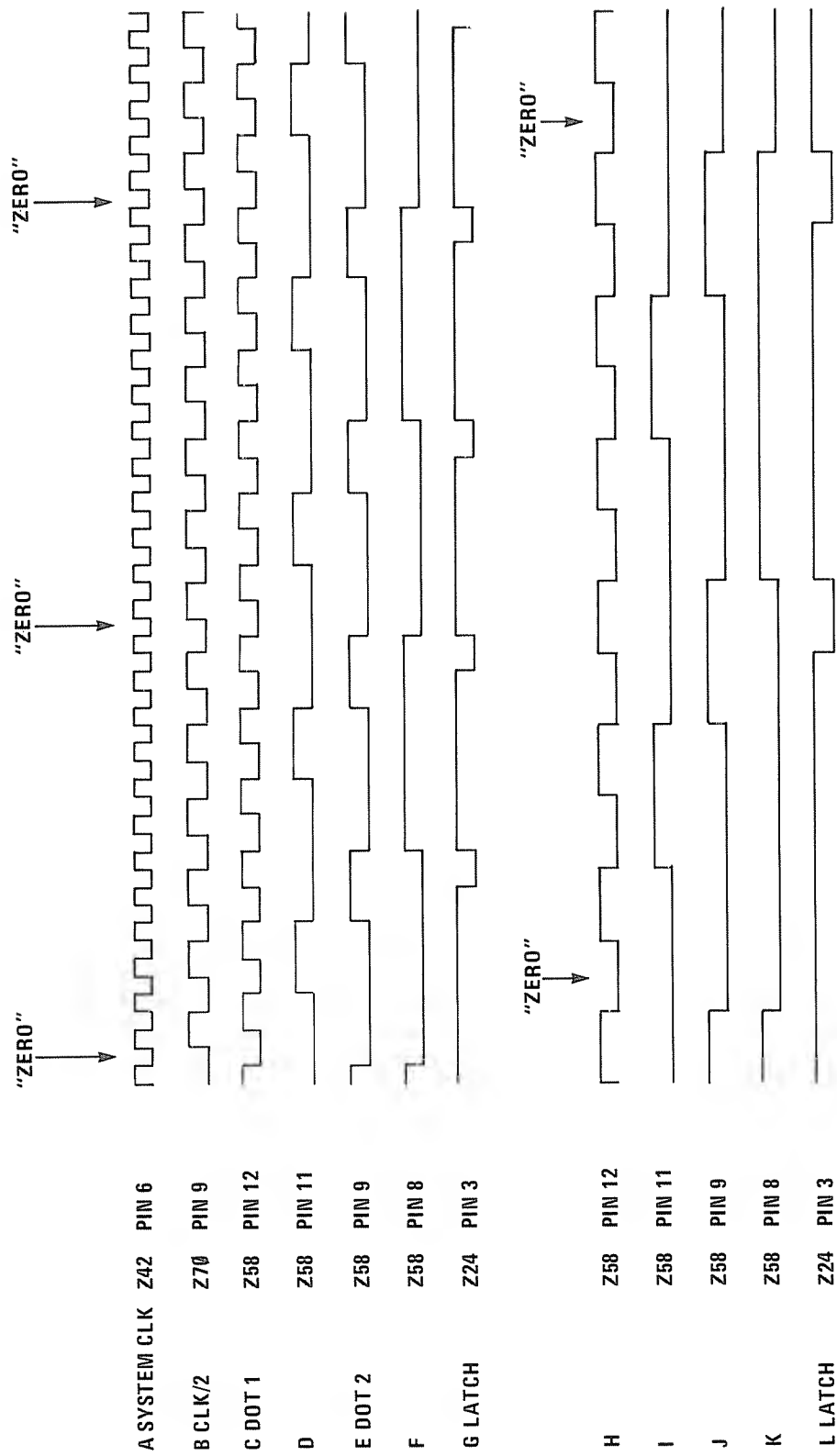


Figure 5. Input Conditioning Waveform

divided by 6 divided by 2 = 887.041 KHz. What did change? Two signals changed. "LATCH" is sourced at Z24, pin 3. In 64 character format, the latch pulse was only one clock cycle (Master Clock) wide, having a period of 6 clock cycles. In 32 character mode, the pulsewidth has doubled to 2 clock cycles and its period is now 12 clock cycles. The other signal that changed was C1. Sourced at Z43, pin 7, it was a square wave at the same rate as the chain signal; but in 32 character mode, it is held low all the time. The signals that changed are very important to the video processor section. The first, LATCH, is used to delay a character between RAM and the character generator. The second signal, C1, determines if the RAM has 1024 or 512 useable addresses.

DIVIDER CHAIN

The divider chain circuit of Z65, Z50, Z12 and Z32 consists of four bit, ripple counters. They have a maximum count of 16, but external circuitry may modify this maximum.

Figure 6 shows a simplified block diagram of the counter chain. Refer to figure 6 and to Sheet 2 during our discussion of the counter chain.

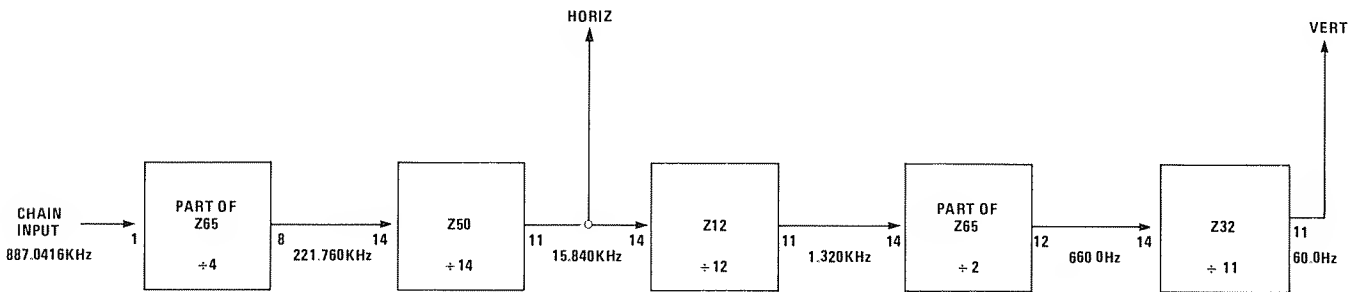


Figure 6. Divider Chain Block Diagram

Upon the next negative transmission of the clock pulse, outputs would look as follows:

Output A = 0
 Output B = 1
 Output C = 1
 Output D = 1

which is equal to 14. But notice AND gate Z66, pins 3, 4, and 5. These pins are tied to outputs B, C, and D. The output of the AND gates pin 6, will go high and clear Z50 back to zero. This clear pulse is extremely rapid – about 50 nanoseconds! The binary count of 14 would therefore be almost invisible to a standard oscilloscope and so would the clear pulse to pins 2 and 3. The time that Z50 is actually reading binary 14 is so short that we can ignore it. Therefore, Z50 will count from 0 to 13 and will then reset back to 0. Since 221.760 KHz is put into Z50, the output at pin 11 will be 15.840 KHz. This frequency will be used by the sync generator circuits to produce horizontal sync.

Z65 is a binary counter that is split into two parts. The chain input from the conditioning logic is applied to pin 1 of Z65. The B and C outputs are used for video RAM addressing, and the output of Z65 at pin 8 is applied to the next counter in the chain. This part of Z65 divides the chain frequency by 4. Since the chain is 887.0461 KHz, the output of Z65, pin 8, is 221.760 KHz. The other part of Z65 will be used later.

The next counter in the chain is Z50. The input is on pin 14 and the divided frequency is at pin 11. This device is externally modified to divide the input frequency by 14. Z50 counts up normally to a binary value of 13. Hence the counter's outputs are as follows:

Pin 12 (Output A) = 1
 Pin 9 (Output B) = 0
 Pin 8 (Output C) = 1
 Pin 11 (Output D) = 1

The next divider is Z12. It is wired to perform a division by 12. It counts up normally until the outputs enable AND gate Z66, pins 9, 10, and 11. This happens at the twelfth falling edge of the clock. Z66, pin 8, will then go high and clear Z12 back to zero. Once again, this clear pulse would be very hard to observe using an oscilloscope. Hence we can ignore this count and consider Z12 as a divide-by-12 counter instead of a divide by 13 counter! If 15.840 KHz is applied to Z12, pin 14, then the output at pin 11 will be 1.32 KHz.

The next divider is part of Z65. On Sheet 2, follow pin 11 of Z12 up to Z65, pin 14. The output is at pin 12. Follow it back down to Z32, pin 14. This part of Z65 divides the 1.32 KHz input by two; therefore, the frequency at pin 14 of Z32 will be 660.0 Hz.

Z32 is the last counter in the chain. It divides the 660 Hz input by 11, producing 60 Hz. Once again, part of Z66 is used to modify the count. When the outputs of Z32 equal

binary 11, Z66 will output a very narrow pulse which clears Z32 back to zero. The 60 Hz at pin 11 is used by the sync generator circuits to produce the vertical sync for the monitor.

VIDEO RAM ADDRESSING

During our discussion of the system block diagram, you noticed that the video RAMs must be addressed from two sections. The CPU must address video RAMs to read or write data from or to specific locations. The divider chain must also address video RAM so that data contained in memory can be processed and displayed on the screen. The video RAMs are addressed either by the CPU or by the divider chain through the use of three multiplexers.

Z64, Z49 and Z31 are the three multiplexers used for video RAM addressing. From the divider chain, there are 10 address lines that will be used to address video RAM. The chain conditioning logic supplies one address — C1. Z65 supplies three addresses — R1, C2 and C4. Z50 supplies three addresses — C8, C16 and C32. Z32 supplies the rest — R2, R4 and R8. Imagine an array of rectangles; 16 rectangles vertically and 64 rectangles horizontally. You would have a total of 1024 rectangles. You could specify any one rectangle by saying, "Starting at the top left hand corner, go down four rows and go to the right 18 columns." The 16 rows could be assigned a binary number from 0 to 15. The 64 columns could be assigned a binary number from 0 to 63. Rectangle 0-0 would be in the upper left hand corner of the array. Rectangle 15-63 would be in the lower right hand corner. Four bits of binary information would therefore specify any one of the 16 rows. It takes six bits of binary data to specify any one of the 64 columns. This is exactly the addressing format used by the counter chain. C1, C2, C4, C8, C16 and C32 specify any column. R1, R2, R4 and R8 specify a row. The row/column addressing format is very useful in troubleshooting video problems in the TRS-80.

The column and row address outputs from the divider chain are applied to the "1" inputs of the multiplexer. Part of the CPU's address bus is tied to the "0" input of the multiplexer. The outputs of the multiplexer are tied to the video RAMs or logic around them. We've got inputs and outputs; how about control? Do you remember the signal VID* that we generated back in the address decoding discussion? We said VID* will select the video RAMs. Notice that pin 1 of the 3 multiplexers is tied to VID*. When the CPU wants control over the video RAM, the address decoder recognizes the video RAM address and causes VID* to go low. When VID* is low, the multiplexer switches the "0" inputs over to the multiplexer outputs. The counter chain addresses are switched out of the circuit, and the CPU has control over video RAM. When VID* goes back high, the CPU is switched out and the counter chain takes over. Most of the time, the counter chain is in control of video RAM. The CPU only takes charge when it needs to. You can see on the display screen when the CPU robs the counter chain of

video RAM control. Ever notice black streaks all over the screen while graphics are being drawn? These streaks are the result of the counter chain losing control over video RAM.

Aside from chain and CPU address, there are inputs to the multiplexer we have not yet mentioned. The first of these inputs is the resistor at pins 13 and 6 of Z49. These two inputs, which are not needed in the counter chain's control over video RAM, are pulled up to 5 volts by R49. Output pins 12 and 7 correspond to the inputs at pins 13 and 6. When the chain has control over video RAM, pins 12 and 7 output a steady state high. Pin 12 goes to the R/W (Read/Write) control of all the RAMs. Since the counter chain never stores data in RAM at the address it specifies, pin 12 should be high when the chain is in control. Pin 7 of Z49 goes to the video RAM data buffer. When the chain is in control, the RAM data bus should be disabled. A high on VRD* (Video Read) guarantees this bus will be off.

We also find WR* and RD* tied to pins 14 and 5 of Z49. When the CPU takes charge of the video RAMs, multiplexer output at pin 12 becomes VWR* (Video Write). The CPU can store data into the video RAMs by causing VWR* to go low. If the CPU wants to read data from video RAMs, RD* can pass through Z49 and activate VRD*. A low here will open data buffers Z60 and Z44. Addressed video RAM data is then placed on the data bus. The CPU can process this data like any other data.

ALPHANUMERIC FORMAT

The CRT (Cathode Ray Tube) in the display will be scanned twice per second. The electronic beam in the CRT travels from top to bottom of the screen and left to right. Each screen or frame consists of 264 scan lines. 192 scan lines are used in the "picture". 72 lines are used during vertical interval and as upper and lower boundaries. Nothing is ever "written" or visible within these 72 lines. There are 1024 character locations per screen (or 512, depending on status of MODESEL). Each character line consists of 64 characters (or 32, depending on the status of MODESEL). There are 16 character lines. Each character line consists of 12 scan lines. An alphanumeric character uses seven scan lines while there are five blank scan lines between character lines. We'll worry about graphics formatting later.

Part of Z65 and Z50 specify the column address. Z32 specifies the row (or character line). Z12 specifies the scan line in any character line. The outputs from Z12 are labeled L1, L2, L4 and L8. These four lines are not used in video RAM addressing because we already stated a row and column address will specify any one of the 1024 rectangles in our rectangle array. Z12's outputs are used in the video processing. L1, L2, and L4 will enable the character generator to output correct data for any character since it knows where the CRT's electron beam is scanning. L8 is used by the video processor to blank (turn off) the five lines between character lines.

Notice NOR gate Z30, pins 8 and 9. These pins enable Z30's output to produce a signal called BLANK*. BLANK* is used by the video processor to give the 72 scan line blanking for the upper and lower boundaries. It also defines the blank boundaries on the left and right of the screen.

VIDEO RAMS

The video RAMs are static and hence do not need refreshing. The data bus is wired in the same way as system RAMs, but with a different enable signal. One interesting point to note: There are seven RAMs. Six are used for ASCII storage, and the seventh is used as a graphic/alphanumeric definition bit. There are eight data lines. Notice the line labeled Bit 6. It is sourced by NOR gate Z30 at pin 13. If Bit 5 (Z62, pin 12) and Bit 7 (Z63, pin 12) are low, then Bit 6 will be high. Z30 is a sneaky way of squeezing a seventh ASCII bit out of six RAMs.

Aside from the data bus, there is another RAM output for data. The video processor needs video data for generation of the alphanumeric and graphic symbols. And this section will be discussed next.

VIDEO PROCESSING

Video Processing consists of five subsections. They are: data latch, character/graphic generator, shift register, sync generator, and video mixing/output driver. The data latch temporarily stores an ASCII or graphic word from video RAM. The latch will retain the byte for processing so that the RAM is free to search out the next byte. The character generator is a ROM that is addressed by the data latch and the scan line signals. This ROM contains the alphanumeric format that makes up all the characters. The graphic generator is not a ROM, but a 4-line-to-1-line data multiplexer. It operates somewhat like a "bit steering" circuit. It steers an ASCII word into a graphic's symbol. The shift register accepts data from the character generator (or the graphic's generator) and converts parallel dot data into serial dot data. Meanwhile, the sync generator circuits have been accepting timing signals from the divider chain. The sync circuits shape up the horizontal and vertical pulses, serrate the vertical interval and sends it all out to video mixing in serial format. In the video mixing section, the serial dot video and the serial sync are brought together. The resulting composite video signal is then "fine tuned" in amplitude and dot-to-sync ratio, and then buffered for a 75 ohm output cable. The signal leaves the TRS-80 and is applied to the display. In the display, the signal is torn apart into its separate components, and you have a readable image on the screen.

DATA LATCH

The data latch consists of Z28 for the ASCII and Z27 for the graphic bit and blanking signals. The inputs of Z28 come from the six video RAMs. The outputs of Z28 go to

Z29, the alphanumeric character generator and to Z8, the graphics generator. The inputs which control Z28 are on pin 9 (latch) and pin 1 (VCLR*). The latch signal at pin 9 is a pulse train developed by the divider chain-input conditioning logic. This signal goes low every six dot cycles (see Figure 5 for latch timing). On the rising edge of latch (low to high transition), ASCII data in RAM is transferred to the outputs and temporarily stored by Z28. RAM data at the input to Z28 may now change, and the RAM has time to search for the next ASCII character. (RAM, any RAM, has a parameter called "Access time". This is the time it takes for the data output to reflect a change after an address change. For example, assume a RAM output is high. The address of the RAM is changed to a new location where a "0" data bit is stored. Even though the RAM is now addressing the low cell, the output still reads the previous high. Only after a short length of time — in the nanoseconds — will the output change from a high to a low.) At the same time Z28 stored the ASCII word, the divider chain changed video RAM addresses. The RAM is now "looking" for the next ASCII word. It has exactly six dot times (about 560 nanoseconds in 65 character format) to find it before the latch is commanded to store the next word.

Z27 is a smaller latch that operates in the exact same manner as Z28. But instead of ASCII, it handles the graphic bit and blanking data. Pin 4 is tied to the inverted output bit from Z63, the graphic RAM. Pin 5 of Z27 has signal BLANK* tied to it. L8 is tied to pin 12 and our "sneaky" bit, bit 6, is tied to pin 13 of Z27. All of these signals are latched into Z27 at the same time as the ASCII word is latched into Z28.

Each input to Z27 has a different function. The graphic bit to pin 4 of Z27 will determine if the ASCII word contained in Z28 is an alphanumeric character or is a graphic word. Pin 5 of Z27, is signal BLANK*. This signal comes from Z30, pin 10, and controls the upper, lower, left and right boundaries of the video display. When BLANK* is high, the CRT's electronic beam is allowed to draw on the screen. When BLANK* is low, the beam is in a boundary area so it prevents the beam from drawing anything. L8 is connected to pin 12 of Z27. L8 acts somewhat like BLANK*. L8, remember, specifies where the electron beam is located in any character line. When low, L8 allows the beam to output alphanumeric dot data. When high, L8 shuts off the beam because it is now scanning one of the five scan lines between character lines. The last piece of data comes into Z27 at pin 13. Sourced at Z30, pin 13, this is the "sneaky bit" that is derived from data contained in RAMs Z63 and Z62. This is the only bit at Z27 which could be considered part of the ASCII word. The output is applied to character generator Z29, at pin 1.

Notice pin 1 of both data latches. This input, when low, will force the latches to their clear state (zero at the outputs). This signal is called VCLR* (Video Clear) and is

sourced by D flip-flop Z7, at pin 6. The flip-flop disables the data latches during a CPU interruption of video RAM. Notice pin 4 of Z7. It is tied to VID*. When VID* goes low, Z7, pin 6, will go low. The low at pin 6 will clear the data latches. (This is what generates the black streaks we discussed in the video RAM addressing section.) When the CPU has finished with video RAM, pin 4 of Z7 goes back high. The next time data is to be latched into Z27 and Z28, Z7 will toggle back to its normal reset state and allow the data latches to operate. If Z7 was not used, we might see characters that appear ripped apart on the screen. For example, assume the CRT was drawing a character when the CPU took command of video RAM. After the CPU finished, the video processing circuit may still see the ASCII code that was in the latch at the time the CPU suddenly jumped in. The video circuit would try to redraw the character on the screen. We would then either see the character twice; or half of it would be over there, and the other half would be here! Clearing out the data latch insures us that the video processor does not get confused.

CHARACTER GENERATOR

Each character consists of a dot matrix. The matrix is five dots wide by seven dots deep. There is one dot between any two adjacent characters that are never turned on. We have five dots, a space, five more dots, a space, etc. Vertical spacing between adjacent data is determined by the frequency of the dot clock. (In the TRS-80, the dot clock signal is labeled SHIFT.) The dot clock is oscillator frequency, in 64 character format, and 1/2 oscillator frequency, in 32 character format. Horizontal spacing between adjacent dots is a function of scan frequency. In other words, each row of dots is aligned along the electron beam's path across the CRT. There are seven rows of character dots and five rows of blanks.

Since each character consists of a pattern of dots, there must be some method to determine which dot should be on and which dot should be off to form any one character. The character generator controls the dot patterns on the screen.

Z29 is the character generator. The seven bit ASCII word, stored in the data latch, is applied to Z29's ASCII inputs, pins 1 through 7. The ASCII addresses a certain area in Z29. You might consider the ASCII inputs to be the higher seven bits of an address. The lower part of the address is inputted at pins 8, 10 and 11. This three bit input selects the row position of the addressed dot pattern. Z29 outputs five dots at one time. Since each character consists of seven rows of five dots, the character generator must output seven separate times just to build one character. Here is how a typical character line is written: Assume an ASCII word is in the latch. The electron beam is on the first scan line of the character. Hence, pins 8, 10 and 11 have a binary "0" applied to them. Z29 outputs the first dot pattern for that particular ASCII character. The next ASCII character is applied to Z29. It outputs the first five

dots for that character. This process goes on until the beam has scanned the entire width of the screen. If we could stop action at this point, all you would have would be a line of dots. On the second scan line, the data at pins 8, 10 and 11 is incremented to read binary "1" (001). The RAM is now prepared to read the second row of dots. The first ASCII character is applied, and it will output the second row of dots for that character. The second ASCII word comes in, and the second row of dots go out. This process continues until all 64 characters have had the second row outputted under the first row of dots. The line counter increments and we apply the first ASCII word once more. We paint a row of dots, increment the line counter and paint another row. Any character in a line is accessed at least seven times. Once the line counter has gone past the seventh count, all the dots make sense; and we will recognize the dot patterns as characters. After the seven dot scans are outputted, the electron beam is turned off; and five rows of blank dots are outputted. We would now be ready to output the first row of dot patterns for the second character line.

The dot output appears slow-reading about it. But ASCII is being shot into the character generator at about a 1.77 MHz rate. The CRT and the retention of the eye make these characters seem like they are outputted whole.

GRAPHICS GENERATOR

Do you remember the rectangle array we discussed in the divider chain section? Well, we are back to the rectangles. As stated earlier, there are 1024 character locations in video RAM. If we divide each large rectangle into six smaller rectangles, we will have the basic graphics cell (Figure 7 shows a divided rectangle). This cell is the smallest piece of graphic information that can be displayed on the screen. Each cell is four scan lines long and three "dots" wide.

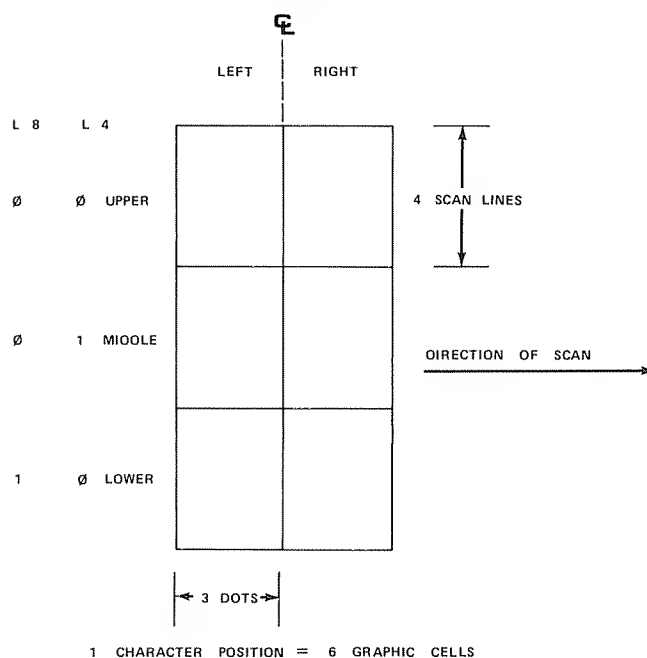


Figure 7. Graphic Cell

Z8 is the graphics generator. Actually, Z8 does not generate anything. Rather, it steers the ASCII addresses around to simulate a graphics generator. The input to Z8 is ASCII from data latch Z28, and the higher order line address from Z12, L4 and L8. L4 and L8 can represent any four numbers from 0 to 3. But since Z12 never goes to binary 12 (except for so short a time we can ignore it), we will only be looking for a binary number from 0 to 3. L8 and L4 are used to specify the vertical address of the six graphic cells. There are three vertical addresses: 00 defines the uppermost pair of cells, 01 defines the middle pair of cells, 10 defines the lower pair. This is also shown in Figure 3.

The ASCII word, labeled LB0 through LB5, determines if the graphics cell is ON (high) or OFF (low). The position of one of these inputs to Z8 determines which side of the center line the cell is located. An input at pin 6 of Z8 specifies a left hand graphic cell. Input at pin 10 of Z8 specifies a right hand graphics cell. Pin 5, left; pin 11, right; pin 4, left; pin 12, right. For example: Assume LB2 is high and all other LB inputs to Z8 are low. LB2 comes into Z8 at pin 5. This pin is associated with a graphic cell location on the left of the character position. Therefore, depending on the status of L8 and L4, LB2 will turn on (light) one of the graphic cells on the left of center line. If L3 and L4 are at logical 00, the upper left cell is turned on. If L3 and L4 are 01, the middle left cell will light.

As you can see, Z8's function as a graphics generator is to steer the ASCII bits around the character rectangle. The vertical position of the graphics in the cell is determined by the status of L8 and L4. The two outputs from Z8 are labeled "left" and "right". This "dot" information is applied to the graphics shift register. It is in shift register logic that data from RAM Z63 determines if graphics or alphanumerics will be written in any one character rectangle.

ALPHANUMERIC/GRAPHIC SHIFT REGISTER

Z10 is the alphanumeric shift register and Z11 is the graphic register. Both devices receive parallel data from their respective generators. The parallel dot data is loaded into the registers and the dot clock (labeled SHIFT) will march the dots out, one behind the other, to the video Mixer. We will discuss the alphanumeric shift register first.

There are a few restrictions when the alphanumeric shift register may serialize dot data and when it shouldn't. First, the data must be alphanumeric and not graphic. Second, the electron beam must be on one of the seven scan lines that are reserved for dot data and not on one of the five lines that are blanked (held off) between character lines. Third, the electron beam must be on one of the 192 scan lines that define the video portion of the screen. (not in boundary space — upper, lower, left or right). Once all three restrictions are met, the dot data is parallel loaded

into the register. NAND gate Z26 insures all conditions are met before data is stored in Z10.

Delay bit 7* is sourced from latch Z27, pin 2, and applied to pin 10 of Z26. When this input is high, data in Z63 is low, which defines an alphanumeric character.

Delay L8 is sourced at Z27, pin 11, and is connected to Z26, pin 12. When this input is high, the beam is scanning in a character line and not between character lines.

Delay BLANK is sourced at pin 7 of latch Z27 and is tied to pin 9 of Z26. When this input is high, the electron beam is in the video portion of the screen and is not located near a sync pulse or in some boundary region.

All three restrictions have then been met. Pin 13 of Z26 is tied to the inverted signal, LATCH. When pin 13 of Z26 goes high, the dot load process will be activated by a low on Z26, pin 8. Upon the next clock pulse at pin 7 of Z10, dot data will be loaded into the shift register. After LATCH goes back high (one dot time after going low), the shift register starts clocking dot data out at pin 13 in a serial stream (when LATCH goes high, pin 13 of Z26 goes low causing pin 8 to go high). Each time LATCH goes high, it forces ASCII and conditional data to be stored in Z27 and Z28. During this time, Z10 will not be shifting dots out at pin 13. Z10 only shifts data out when pin 15 is high. When low, pin 15 forces Z10 to load data from the character generator.

Notice the unused inputs to Z10. Pin 9, the clear input, is pulled up, via R40, to VCC. When this pin is low (due to a short), you would have a blank screen. Pins 14, 3, 2, 1 and 6 are tied to ground. Pin 14 gives you that blank dot between adjacent characters. Pins 2 and 3 are not used, but are register inputs like pin 4 or 5. Pin 1 is for serial data input and pin 6 inhibits the clock input. They are not necessary in this application.

The graphics shift register is Z11. Operation is almost the same as Z10, except for the condition that must be met for use. The graphics conditions are as follows: First, Z63 must specify a graphics character instead of an alphanumeric character. Second, the electron beam must be in the video region of the screen. There are only two conditions restricting graphics. Since a character rectangle ends where another starts, there is no inter-character line blanking. If you turn on all of the graphic cells, you would have a full, large square with no holes and boundaries surrounding the square. Once all of the restrictions are met, graphic dot data may be loaded into Z11 for shifting to the video mixer. The other NAND gate in Z26 is used as the graphics load enable.

The inverse of delay bit 7* (DLY bit 7) is sourced at Z27, pin 3. It is applied to Z26, pin 4. When high, this input tells Z26 that Z63 does indeed contain a "1" which defines a graphic code.

Delay Blank (DLY BLANK) is tied to Z26, pins 1 and 2. When high, this input tells Z26 that the electron beam is indeed in the video portion of the screen.

Once all conditions are met and LATCH goes low, Z26 will go low. Just like Z10, Z11 will load dot data; and when pin 15 goes back high, the shift process will start. The six graphic dots are shifted out of pin 13.

Notice pin 9 of Z11 is pulled up by R40. Likewise, pins 3, 2, 1 and 6 are tied to ground. But pin 14 is used this time. In graphics, there is not a blank dot space between character rectangles.

SYNC GENERATOR

The Sync Generator circuit accepts timing signals from the divider chain to develop horizontal and vertical sync pulses for the display. These pulses are used by the display to control the CRT's electron beam.

The sync pulses are generated by logic which operate like linear elements.

Z6, a CMOS inverter, is used to generate the horizontal pulse; and Z57 is used to generate the vertical pulse. Signal HDRV (Horizontal Drive) is sourced from the divider chain at Z50, pin 11. This signal is buffered by Z6, pins 13, 12, 1 and 2, and applied to potentiometer R20. R20 controls where the horizontal pulse starts in reference to HDRV. When R20's wiper is close to Z6, pin 2, the horizontal pulse will start almost at the same time as HDRV goes high. When the wiper is moved in the opposite direction, there is a delay between the time HDRV goes high and the time the horizontal pulse starts. R20 is not performing this phase shift by itself. C20, together with two inverters in Z6, form the complete shift network.

Here's how it works: HDRV goes high, causing Z6, pin 2, to go high (in this case about 5.0 volts). A current flows through R20 charging C20. While C20 charges, the voltage at pin 3 of Z6 slowly increases from zero as the current through R20 decreases. After a length of time, the voltage at pin 3 of Z6 will be high enough for pin 3 to "see" a high. Z6, pin 4, goes low causing pin 6 to go high. C20 rapidly charges. Everything stays in this mode until HDRV goes low. At this point, C20 starts to discharge at the same rate it charged. When the voltage at Z6, pin 3, decreases to a logical "0" level, pin 4 will go high, causing pin 6 to go low. C20 rapidly discharges. The process cycle is now complete until the next time HDRV goes high. The time the voltage level at pin 3 of Z6 stays above the minimum logical "1" level determines the amount of shift from HDRV. The effect of R20's position, which adjusts the delay time, on the screen is a horizontal shift of video display.

After the horizontal signal is phase shifted, the horizontal pulse must be shaped. C21 and R43 form a differentiation network which creates a smaller pulse of known width from the shifted HDRV signal. Operation is quite simple. When Z6, pin 6, goes high, C21 and R43 differentiate the rising edge. A narrow pulse is passed to Z6, pin 11, inverted by pin 10 and inverted and buffered by Z6, pins 9 and 8. The next result is a pulse about four microseconds long, called horizontal sync.

The vertical sync phase shift operates in the exact same manner as the horizontal. Z57 is used as the active element which R21 and C26 form the delay network. The differential network consists of C27 and R44. Notice the only difference between horizontal and vertical circuits is the value of the two capacitive devices.

HORIZONTAL AND VERTICAL MIXING

Once the two sync pulses are phase shifted and pulse-shaped, NAND gate Z5 is used to mix the two signals together and serrate the vertical interval. Figure 8 shows idealized waveforms around Z5.

At line A, the horizontal pulses are shown. The source for this output is Z6, pin 8. At line B, the vertical pulse is shown coming from Z57, pin 8. Z5, pins 1 and 2 are tied to the waveforms shown at lines A and B, and the resulting NANDed output is shown on line C. Line C in Figure 8 is now used as a source to NAND the horizontal and vertical syncs once more. Line D shows the result of NANDing line C with line A. Line E shows the result of NANDing Line C with Line B. Lines D and E are NANDed by Z5, pins 10 and 9. The resulting mixed sync waveshape is shown on Line F. Notice two things about Z5. First, pin 8's output shown in Figure 8, line F, is "false" composite sync. In other words, it is inverted away from true form. Secondly, notice Z5 may be evaluated down using Boolean algebra into a 2-input exclusive OR gate. The output at line C may be expressed as $V\bar{H} + H\bar{V}$, where V is vertical sync at Line B and H is horizontal sync at line A in Figure 8.

VIDEO MIXING

The video mixing circuitry generates the composite video signal for the display. The video mixer accepts both alphanumeric or graphics dot data from the shift register, level-shifts it, and places it atop the composite syncs. The composite waveform is then buffered to drive a 75 ohm impedance and is sent, via cable, to the video display.

Dot data from shift register Z10 or Z11 is applied to Z30, pin 3 or pin 2. You should never see both pin 3 and pin 2 active at the same time. While Z10 is outputting alphanumeric data, Z11, pin 13, should be low. Conversely, if Z11 is outputting graphic data, Z10, pin 13, should be low. The net result at pin 1, Z30, is a single waveshape of video dot data. This data is applied to Z41, pins 6 and 7.

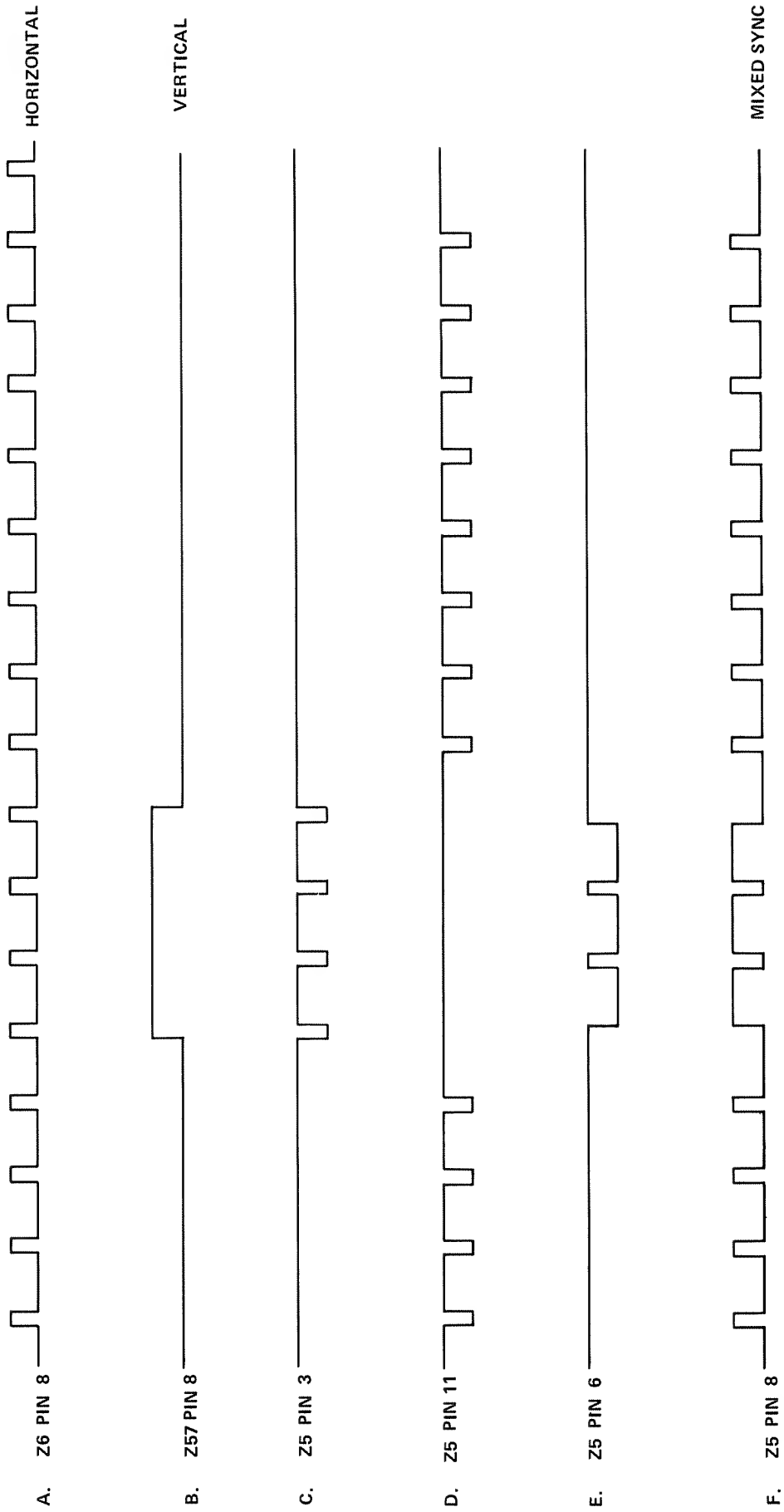


Figure 8. Sync Mixing

The composite sync data is sourced from Z5, pin 8, and is applied to the base of transistor Q2. Each time the base of Q2 goes about 0.6 volts below 5 volts, Q2 turns on, which applies 5 volts to resistor R28. (Actually, the voltage applied to R28 will be less than 5.0 volts, due to the saturation voltage of Q2.)

The dot data from Z30, pin 1, is inverted by Z41, pins 6 and 7. The resulting output at pin 5 is a normally low signal which goes high only when the shift registers output a dot. Z41 is a high current driver. The output at pin 5 is the collector of the output buffer transistors. So, essentially, we have the video and sync going to two transistors. These transistors act as switches controlling current flow through resistor network R28, R27 and R23. Figure 9A shows a simplified drawing of the circuit.

Q2 and Z41 are represented as switches. When Q2 is "opened", there is no voltage applied to R28; and the output node is at ground level. When Q2 "closes" and with Z41 also held "closed", the output voltage goes up to about 1.23 volts. This voltage will be called "the black level". Voltage below this level is "blacker-than-black" and is known as sync level. Voltage above 1.23 volts can be called the "white level". Normally, the black level stays at 1.23 until the sync at Z5, pin 8, goes high, turning off Q2, forcing the output at the node to go to ground. When dot data causes switch Z41 to open, the voltage at the output node increases to about 2.75 volts.

We now have a signal at the output node that contains both video dots and sync information. This signal is almost ready for the display. All that is necessary at this point is a little level shifting and output buffering.

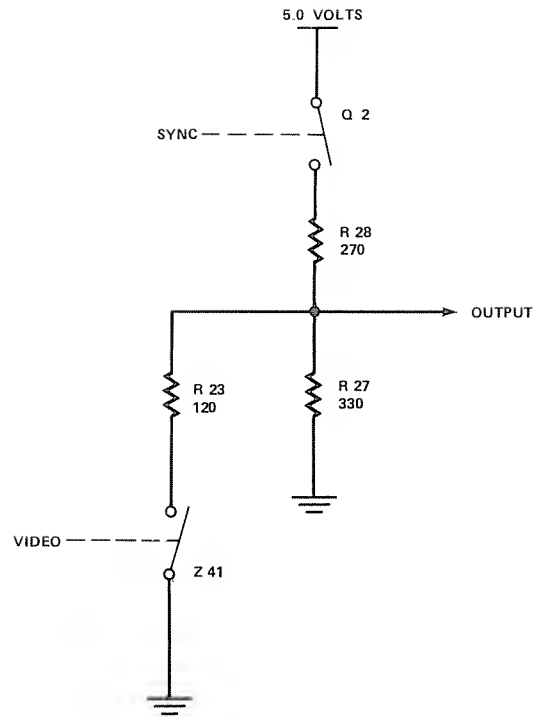


Figure 9A. Simplified Video Mixing

Transistor Q1 is used as a common emitter amplifier. Composite video is applied to the base of Q1; and the emitter outputs the waveform shown in Figure 9B. This final signal is used by the terminal for operation. Capacitors C7 and C2, together with R30, form a filter network for Q1's collector. The capacitors ensure the DC bias level on the collector is video free, and helps in reducing power dissipation in Q1.

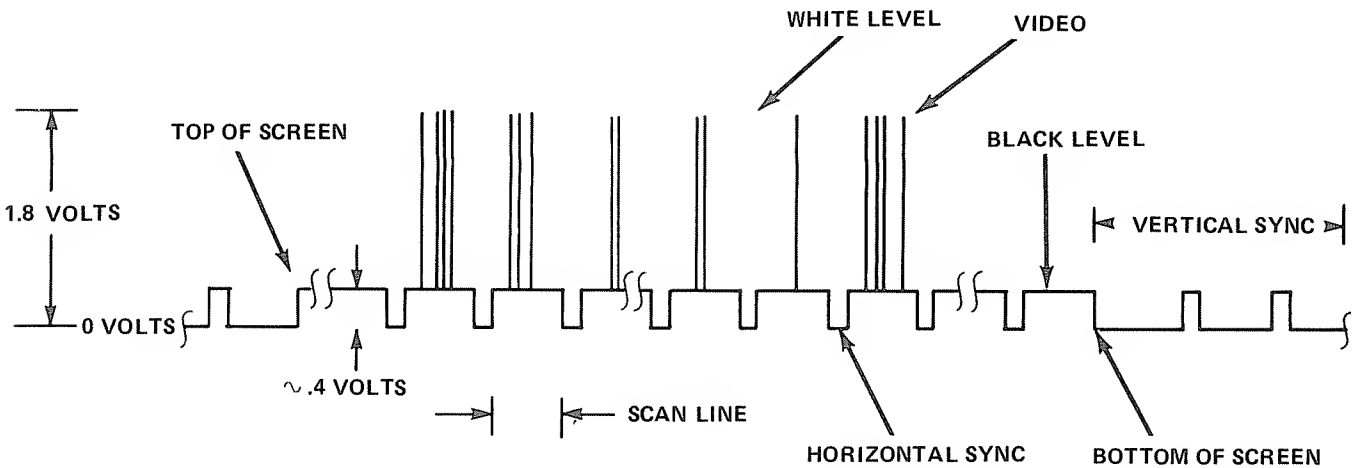


Figure 9B. Composite Video Output

KEYBOARD

The TRS-80 keyboard consists of 53 single pole, single throw normally open keys molded in a plastic base. The base is mounted, together with four ICs and associated resistors, to the keyboard PCB. As you can see from the schematic, this keyboard does not output ASCII. It is scanned not unlike calculator type keyboards. Each key represents a switch across a matrix node. When closed, the switch will short out a horizontal line to a vertical line. ROM software will detect the node short and generate ASCII equivalents for that particular key.

The keyboard is accessed by decoder signal KYBD*. When this signal goes low, it enables tristate buffers Z3 and Z4. The inputs to these buffers are normally held high by the pull-up resistors at the top of the keyboard schematic, R1 through R8. All of the horizontal address lines are made to go high at the same time as KYBD* goes low. If the CPU detects a logical "1" on one of the data lines, there is a key pressed on the keyboard. The CPU ROM will then scan the address lines one-by-one until it finds the "1" output on the data bus again. After finding it, the ROM can instruct the CPU how to generate the ASCII code for that particular key. At this time, the CPU also checks the status of the two shift keys. If one of these keys is not pressed, the ASCII code is not modified. If a shift key is pressed, the ASCII is modified accordingly.

Only one point should be brought up about the keyboard. The inverters on the address lines are open collector types. You may not be able to see the address signal on Z1 or Z2's output unless one of the keys associated with that output is pressed. With no key pressed, there is no voltage applied to the KR (Keyboard Row) lines. When a key is pressed, the associated pull-up resistor supplies voltage. Then you will be able to see activity on a KR line.

INPUT AND OUTPUT PORT

The TRS-80 microcomputer system is memory mapped. But it does have input/output ports. The basic difference in memory mapping and ports is in the method data is handled. In memory mapping, the CPU knows where the data is. In a port, the CPU does not know and couldn't care where the data is located. If the port is some kind of memory, the CPU will output data to that port; and it would be up to port circuitry to process and store data. In the input condition, the CPU accesses the input port; and it is up to the port to find data and feed this data to the data bus for the CPU.

The Z-80 CPU can access up to 256 output/input ports. In the TRS-80 system, we only use one. The cassette recorder is the only port used. Its address is FF in hex. (Ports are accessed using only the lower eight address lines.)

PORT ADDRESSING

Since the TRS-80 uses only one output/input port, there must be only one port decoder. The port decoder is shown on Sheet 2, between the sync mixing circuit and the power supply.

Z54 monitors address bits Z1 through Z7. Z52, pin 5, monitors the A0 line. When hex FF is outputted on address lines A0 through A8, Z54 pin 8, and Z52 pin 6, will go low. These two outputs are tied to OR gate Z36, pins 2 and 1. When A0* and FE* are low, FF* at pin 3 of Z36 will go low. The port address decoding is now complete. If we have a low at OUT* (the CPU wants to access an output port when this signal is low), Z25, pin 8, will output OUTSIG* because OUT* and FF* are low. If we have a low at IN* (the CPU wants to access an input port), Z25, pin 6, will output a low generating INSIG* because IN* and FF* are low. IN* and OUT* should never be active (low) at the same time. Hence, INSIG* and OUTSIG* should never be active (low) at the same time.

OUTSIG*

The OUTSIG* line is used to control two cassette functions and one video function. It is used to generate the audio signal for the cassette recorder in a CSAVE condition. It is used to control the recorder's motor also. Its video function is to control signal MODESEL (Mode Select). MODESEL will change 64 character format to 32 character format or vice versa. OUTSIG* is also controlling a latch made up of NAND gates in Z24. We will discuss this circuit later.

Z59 is a data latch controlled by OUTSIG*. This latch accepts data on lines D0 through D3. D0 and D1 are tied to pins 4 and 5 of Z59. These two inputs are used to input data that is recorded on tape during a CSAVE function. D2 is connected to pin 12 of Z59. This input controls the status of the recorder's motor. D3 is connected to pin 18 of Z59. This input controls the status of MODESEL*.

The input to latch Z59 is stored and transferred to the output each time OUTSIG* goes high (rising edge triggered). For example, if input D2 is high when OUTSIG* goes high, pin 10 of Z59 will go high and stay high. The recorder's motor will turn on. If input D2 is low when OUTSIG* goes high, pin 10 of Z59 will go low and the recorder's motor will be off.

CASSETTE MOTOR CONTROL

At the start of a CSAVE function, the cassette recorder motor must be turned on. The CPU will cause OUTSIG* to go low and apply a logical "1" to D2. When OUTSIG* goes high, the high on D2 will be transferred and held at pin 10 of Z59. This output is connected to relay drive Z41, pins

1 and 2. Pin 3 of Z41 will go low causing current to flow through relay K1's coil. K1's contacts close, shorting out pins 1 and 3 of J3. These two pins are associated with the remote jack at the recorder. The recorder's motor will then turn on.

Notice diode CR3 and Zeners CR9 and CR10. CR3 is a standard silicon diode used for an "anti-chatter" function. When power is applied or removed from K1's coil, a counter EMF is generated. This voltage could be high enough to damage Z41's output transistor, or cause K1 to click off and on a couple of times producing undue wear to the switching contacts. CR3 will shunt the counter EMF voltage around K1 and prevent transistor damage or relay chatter. Zener diodes CR9 and CR10 are used in somewhat the same way. Here we're trying to protect K1's switch contacts. When the recorder is turned on, a high voltage

spike could be produced. It could be deadly – the contacts could weld together because of a high voltage arc. CR6 and CR10 prevent possible damage by shunting any voltage spikes above a certain level away from the contacts.

CASSETTE AUDIO OUTPUT

After the motor is turned on, the CPU may output data for storage on tape. All data timing for this output function is software controlled. Z59 is used to store data from the CPU and it "builds" the output waveform using CPU data. CPU data, under software control, is applied to latch Z59 on pins 4 and 5. Output pins 2 and 6 are connected to a resistor network consisting of R53 through R56. As OUTSIG* is clocking data into Z59, the resulting output on the line labeled CASSOUT, resembles a sine wave built out of square waves. Figure 10 is an illustration of one bit time.

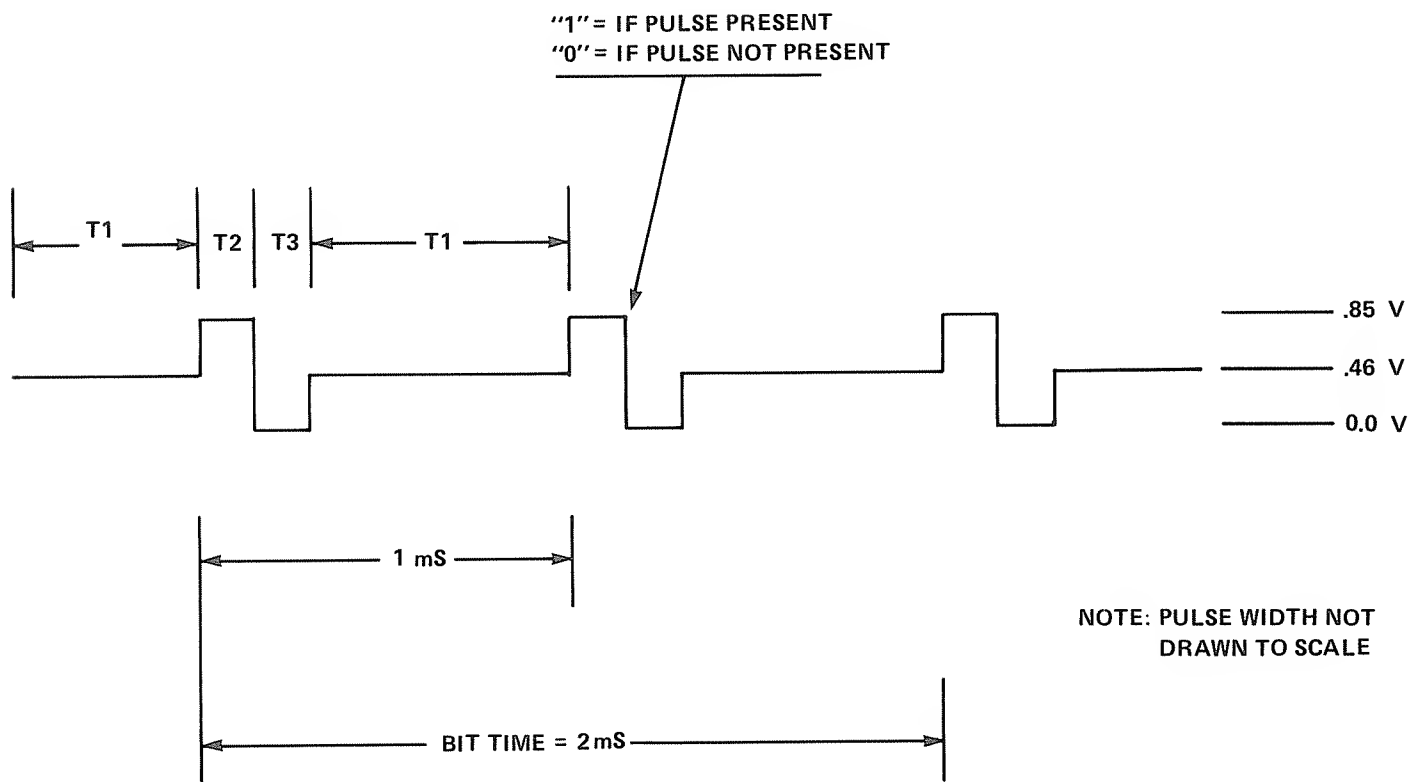


Figure 10. Idealized CASSOUT

In Figure 10, the voltage output is a function of the status of pins 2 and 6 of Z59. In the period labeled T1, the output is shown as .46 volt. T1 is when output pin 2 is zero and output pin 6 is high (D0=0; D1=0). The voltage during T2 is outputted when pin 2 is high and pin 6 is high (D0=1; D1=0). The voltage during T3 is outputted when pin 2 is low and pin 6 is low (D0=0; D1=1). All "digital sine waves" are produced in this way.

Notice the time periods shown in Figure 10. From the start of one bit time to the start of the next bit time is two milliseconds. A one or zero is dependent upon the presence or absence of a pulse between the start of two bit times. For example, when the CPU outputs a one bit, it will generate a start pulse. One millisecond later, another pulse

will be generated. One millisecond later, the start pulse of a new bit is generated. If this bit is to be a zero, then there will be a two millisecond delay before another pulse is generated; and this pulse starts the third bit time. The pulses are outputted to the cassette recorder from pin 5 of J3. This pin is tied to the AUX input of the recorder. The CPU outputs all of the instructions in system RAM to tape during the CSAVE function. When the function is complete, audio to the recorder is disabled and a low is outputted at D2, shutting off the recorder's motor.

Data is written on the tape in the following format: Upon CSAVE, the CPU forces Z59 to output 128 zero bits. It then outputs hex code A5 used by the CPU during CLOAD for synchronization. A two byte starting address and a two byte ending address is added next. Then the data follows, however long it is. After the data, the last portion to be stored on tape is the check sum. This one byte number is the sum of all data added together. It is used by the CPU to ensure what it CLOADed-in is what was CSAVEd-out. If the check sums don't match up, then there was a load error.

CASSETTE AUDIO INPUT

If the recorder could faithfully give back what was sent to it, we could eliminate a quad operational amplifier and a handful of passive components. But, it doesn't; so Z4 stays

in. Matter of fact, the recorder adds stuff to the signal. Motor noise and 60 cycle hum complicate signal processing considerably.

Upon a CLOAD instruction from the CPU, the recorder motor turns on and cassette audio is applied to pin 4 of J3. This signal called CASSIN, is tied to capacitor C24 and resistor R67 at the input of the audio processor section. Z4, pins 1, 6 and the output pin 5, form an active filter. This part of the circuit is used to filter out undesired noise and hum from CASSIN. It is a high pass filter, with about a 2 KHz roll off.

If you looked at CASSIN using an oscilloscope, you would see the data pulses riding atop a 60 Hz hum signal. After passing through the high pass filter, the resulting waveform would have the 60 Hz hum removed and only the data pulses would be left. The signals are swinging above and below a base line of about 2.0V. Figure 11 shows some idealized cassette signals. The signal drawn at line A is the type that could be expected at the output of the active filter.

Once filtered, the next section of Z4 is used as an active rectifier. CR4 and CR5, together with the biasing resistors around pins 2, 3, and 4 will full-wave rectify the data pulses. A typical output on the cathode side of CR4 is shown on line B of Figure 11.

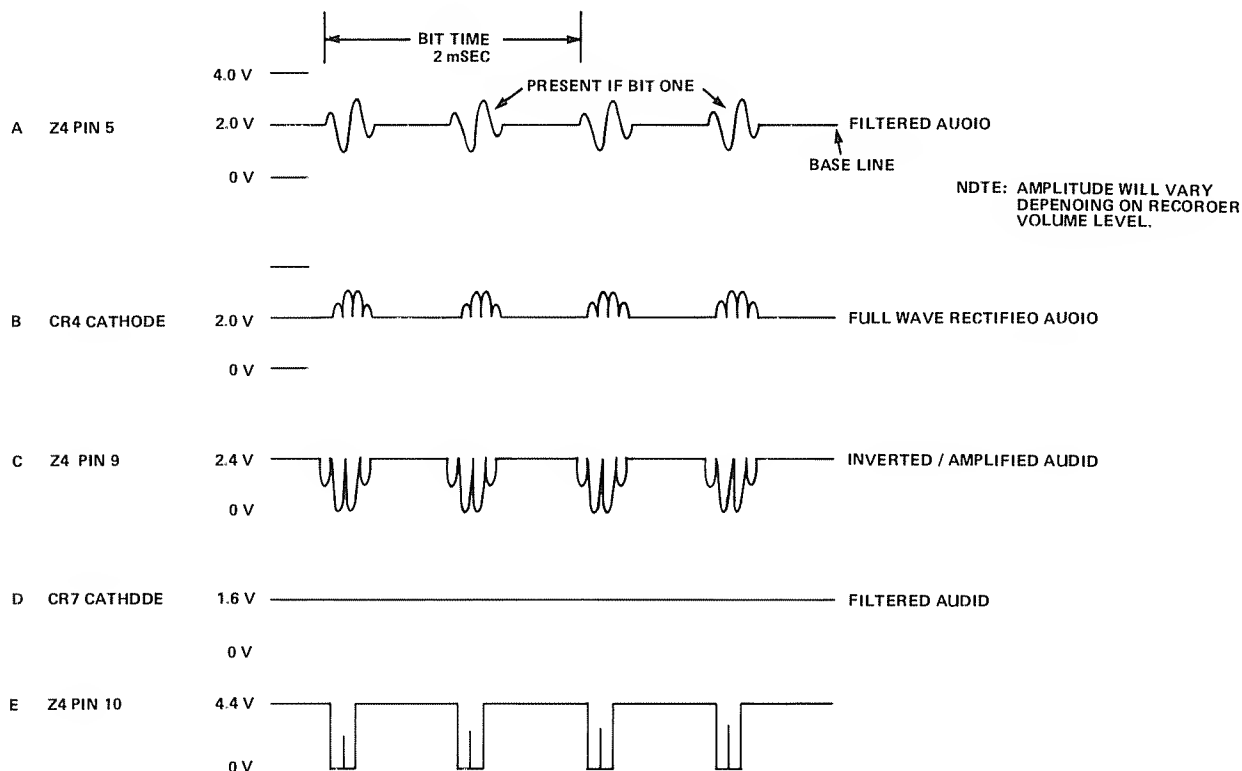


Figure 11. Audio Processing

After rectification, the signal is inverted and amplified. Z4, pins 8, 13 and 9, is wired to form an inverting amplifier circuit. The ratio of R41 to R42 gives the amplifier a gain of about 2. Line C in Figure 4 shows a typical output at Z4, pin 9.

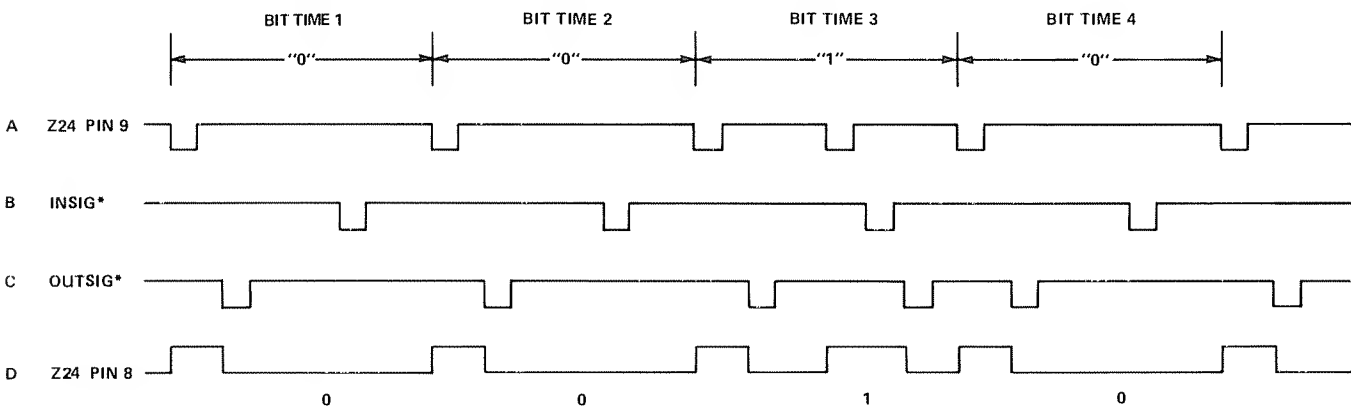
The last stage of Z4 is used as a level detector. CR6 and CR7, together with C39, form a power supply of sorts. The amplified audio signal from Z4, pin 9, is applied to the anode of CR6. CR6 and CR7 decrease the voltage level of the incoming signal by about .8 to 1.0 volt. C39 filters the resulting voltage and creates a DC signal like the one shown on line D of Figure 11. If the signal output from Z4, pin 9, drops below the reference voltage level at C39, Z4, pin 10, will go low. It will stay low as long as the voltage on pin 12 of Z4 stays below the reference. Line E shows the resulting output from Z4, pin 10. Notice that we lost a couple of pulses of audio because the signal did not swing toward ground enough to trigger Z4, pin 10. The negative transaction at pin 10 will be used to set flip-flop Z24. Cassette data will be converted into program data by the software in ROM and the CPU.

INSIG*

Exactly how the CPU turns a string of ones and zeroes into the text of a basic program would interest only the hardcore software person. The amount of hardware used in the TRS-80 to get cassette data to the CPU is minimal. Only the hardware will be discussed.

Z25, pin 4, is tied to IN*. This signal will go low when the CPU wants to input data from a port. Port addressing has already been discussed. A low at pin 4 of Z25 and a low at address decoder Z36, pin 3, will cause a low at Z25, pin 6, INSIG*. This signal is controlling only one device — part of Z44. Z44, pin 12, is tied to pin 8 of NAND gate Z24. The two NAND gates of Z24 are wired to form a set-reset latch.

If pin 9 goes low, pin 8 will go high. Pin 8 is cross tied to pin 12. If pin 13 is high and since 12 is high, pin 11 will be low. With a high at pin 8 and a low at pin 11, the flip-flop is said to be set. If pin 8 is low and pin 11 is high, the flip-flop is reset. The flip-flop is being set by cassette data and reset by OUTSIG*. Z44 monitors the status of Z24 under command from INSIG*. Here is how it works during a CLOAD function: When CLOAD is entered via the keyboard, OUTSIG* will go low, starting the recorder's motor and resetting Z24 by pulsing pin 13 low. The first time Z24, pin 9, goes low, starts the first bit time. This is shown in Figure 12 at line A. Line D, the output of the latch, goes high as soon as pin 9 goes low. OUTSIG* goes low after a short time delay, shown on line C. This signal will reset the flip-flop as line D shows. A short time after OUTSIG* goes back high, the CPU will test Z24, pin 8's status by enabling Z44. Line D is low at this time. The CPU recognizes a logical "0" during bit time 1 as shown by the 0 under line D. The next time line A goes low is the start of bit time 2. The low on Z24, pin 9, sets the flip-flop. OUTSIG* resets the flip-flop a short time later. INSIG* then enables Z44 and checks the status of the flip-flop. The CPU "sees" a zero again, so bit time 2 is a zero bit. The next low on line A starts bit time 3. It sets the flip-flop, and a short time later OUTSIG* resets the flip-flop. Before INSIG* can test status, another low comes from the audio processing level detector and sets the flip-flop. Now INSIG* goes low, checking status. It finds Z24, pin 8, high. The CPU labels bit time 3 a "1". Now the CPU must reset the flip-flop before bit time 4 starts. Line C shows the added OUTSIG* pulse to reset Z24. The flip-flop is reset and stays reset until the next low on line A sets it again. The CPU finds bit time 4 to contain a zero. This set/reset process continues until the CPU has read every bit time of the program that was stored in the cassette. It is the CPU's responsibility to assemble the bit times into data words; the words into text; and store the text in RAM. The CPU is quite busy during a CLOAD function.



NOTE: PULSE WIDTH NOT DRAWN TO SCALE

Figure 12. Data Latch Timing

SYSTEM POWER SUPPLY

The TRS-80 needs three voltage levels: +12 volts at about 350 milliamps; +5 volts at about 1.2 amps; and -5 volts at 1 milliamp. The +12 and -5 volts are needed by system RAM and everything needs +5 volts. The +12 volt and +5 volt supplies are regulated and current protected against shorts. The -5 volts supply is not as critical as the other two supplies, and it uses a single zener diode for regulation. Raw, unregulated power is supplied to all regulator circuits from a UL approved "AC adapter."

AC ADAPTER

The AC adapter (or power pack) is a large version of the type used in calculators or TV game products. Inside the plastic case is a single transformer with one primary and two secondary windings. The primary circuit is designed for 115 VAC and has an operating range of 105 to 135 VAC. There is a wire fuse in the primary side to meet UL specifications.

The two secondary circuits are both center tapped. One secondary is rated at 14 volts AC at 1 amp. This circuit is used in the +5 and -5 volt supplies. The other secondary winding uses internal diodes and it outputs 19.8 VDC at about 350 milliamps. This circuit is used in generating the 12 volt supply. All voltage outputs and center taps are brought into the power input at J1.

+12 V POWER SUPPLY

Raw, unregulated voltage for the +12V supply is inputted at pin 2 of J1. When power switch S1 is closed, C8 filters the voltage and the net result is 20 volts, or so, applied to Q6 and to regulator Z2. Figure 13 shows a simplified diagram of the internal circuitry in a 723 regulator chip. The Figure will help in the regulator operation discussion.

The filtered DC voltage from the power pack and C8 is applied to pin 12 of Z2 and the emitter of series pass transistor Q6. The voltage applied to pin 12 allows a constant current source to supply zener current for Za. Pin 6 of Z2 will output a zener voltage of about 7.15 volts. Pin 6 is tied to pin 5, the positive input to operational amplifier Zb. The negative input to the op-amp is tied to the wiper of R10. Initially, pin 4 of Z2 is at ground, forcing the output of op-amp Zb to output about 7.15 volts. Transistor Qa turns on which turns on pass transistor Q6. The pass transistor supplies voltage for current monitoring resistor R18 and to the resistor network R13, R10 and R12. If R10 is adjusted for 7.15 volts at its wiper, the op-amp will be balanced and Q6 will output only enough voltage to keep the loop stable. If output voltage decreased below 12 volts, Zb's output would decrease which would force the current through Qa to decrease. Qa would cause Q6 to increase the current through it, and the output would rise back up to the 12 volt level. If the 12 volt line increased in voltage, the op-amp would cause Qa's current to increase, forcing Q6 to slow down.

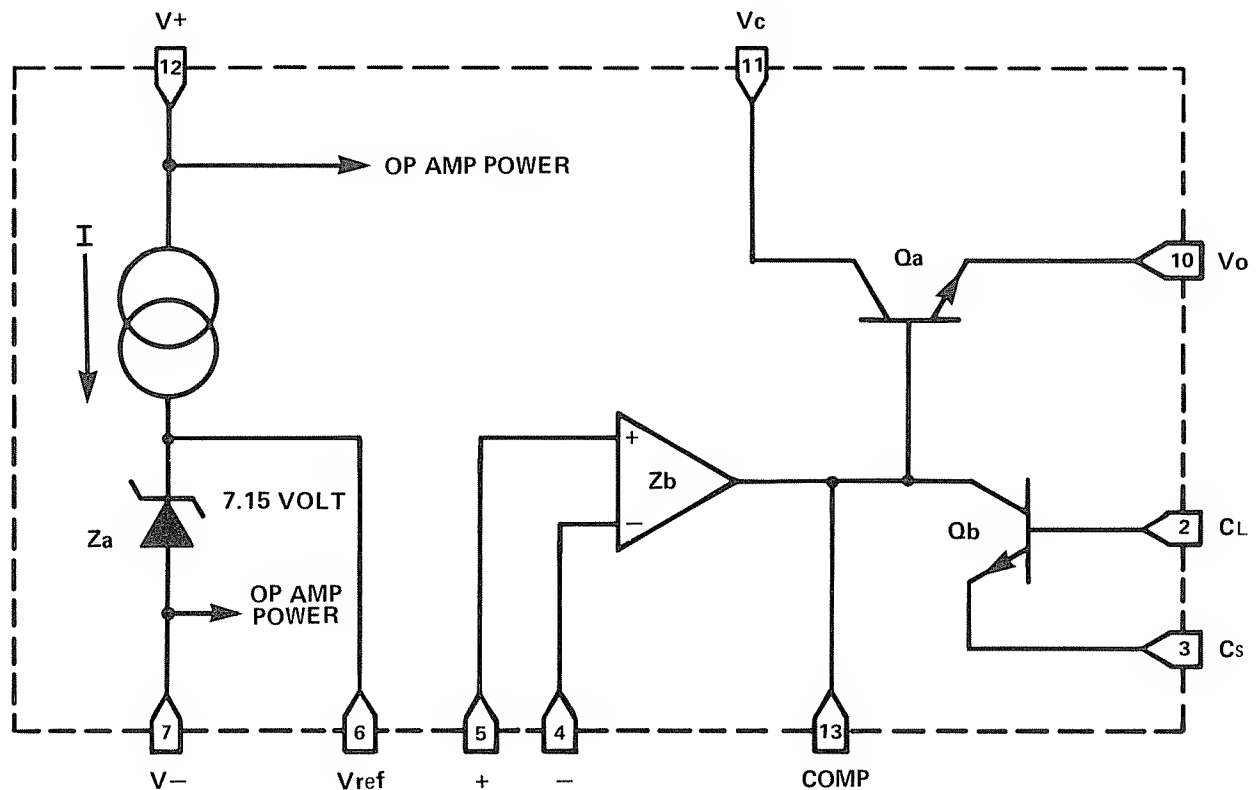


Figure 13. Block Diagram of 723 Regulator

The transistor labeled Qb, in Figure 13, is used to protect power transistor Q6 against over-current damage. If R18 drops sufficient voltage to cause the resistor node at Z2, pin 2, to reach 12.6 volts, Qb will take command of Qa. As Qb is turned on, Qa turns off which starts turning Q6 off. The voltage at Z2, pin 10, must approach 14.7 volts before Qb takes charge of Qa. 14.7 volts at pin 10 means that the 12 volt supply is approaching its maximum design current of 480 milliamps. If a short develops across the 12 V supply, Qb will activate, forcing Qa to shut down. With Qa off, Q6's base rises to the input voltage level because of R16. Q6 snaps off the supply, preventing it from attempting thermal suicide. Once the short is removed, Qb will turn off and the system will operate normally.

Capacitor C13 connected between pins 13 and 4 is a frequency compensation capacitor. It prevents the op-amp loop from going into oscillation. C11 and C15 are the supply's output filter and noise suppressors. Capacitors C28, C30, C32 and C34 are distributed along the 12 volt supply bus for transistor suppression.

+5 VOLT SUPPLY

The 5 volt power supply also uses a 723 regulator. Due to the current and voltage requirements, more components were stuck around the regulator for support. But the basic circuit operates the same. Figure 13 will also be used in this circuit.

For the 5 volt supply, the AC adapter supplies about 17 volts AC at J1, pins 1 and 3. Full-wave rectifier, CR8, rectifies the AC. When S1 is closed, about 7 VDC is passed through the switch contacts and is filtered by C9.

The power supply for Z1 and the current source for zener Za is taken from the regulated side of R18 in the 12 volt section. Pin 7 is grounded as in Z2, but the zener output is handled differently. The 7.15 volt zener voltage is applied to the resistor network consisting of R6, R5 and R11. When R5 has been adjusted for a 5 volt output on the supply bus, pin 5 of Z1 will be at about 5 volts. The negative input of the op-amp, Zb, is sourced through a 1.2K resistor, R7, and tied to the 5 volt bus. The op-amp controls Qa, which controls base drive for Q3. Q3 is used to handle the greater base drive necessary for pass transistor Q4. Q4's collector is tied to current sensing resistor, R4. R4 monitors the current the 5 volt bus is producing just as R18 did for the 12 volt bus.

Circuit operation is exactly the same for Z1 as it was for Z2. If op-amp Z6 detects a rising or falling voltage condition at the output bus, it will adjust base current to Qa. Since Qa cannot handle the drive requirements for Q4 directly, Q3 is needed for current gain. During current limiting condition, Q5 monitors the voltage across R4, which is a direct function of bus current. As Q5 begins to

turn on, the node at R3 and R9 supplies more voltage for base drive of Qb. As Qb takes command of the regulator loop, Qa is commanded to start cutting Q3 off. Q3 begins to turn Q4 off and the circuit goes into current limiting. The current limiting action of Q5 starts to come into play when the voltage across R4 approaches .6 volt. Ohm's Law tells us the bus current at this voltage level is approaching 1.82 amps.

C12, connected between pins 13 and 14 of Z1, performs the same compensation function as C13. C10 and C14 are the output filter and noise suppressors while the thirty-two .01 microfarad capacitors are distributed all over the Board to suppress transit spikes. Notice zener diode CR1 on the 5 volt bus. This diode is used as crowbar circuit protection in case of catastrophic failure in the RAMs. If something happens in the system RAM circuit that causes a short between the 12 and 5 volt buses, CR1 would turn on, causing the 5 volt bus to go into current limiting. Since CR2 is a 6.2 volt zener, it would protect the TTL devices connected to the 5 volt bus from being damaged by a sudden 12 volt supply voltage. Normally, CR1 would be off with no current flowing through it.

Notice one item: The 12 volt supply must be working properly before the 5 volt supply will operate correctly. Therefore, the 12 volt supply must be adjusted before the 5 volt supply.

-5 VOLT SUPPLY

Source voltage for the -5 volt supply comes from the negative terminal of rectifier CR8. When switch S1 is closed, the negative DC is filtered by C1 and about -11 volts is applied to resistor R19. R19 is used to limit current for zener regulator CR2, a 5.1 volt device. The -5 volt circuit is about as simple a power supply as can be designed. C4 and C3 is the -5 volt supply filtering and noise suppressing caps while C16 through C19 perform the transit suppression function.

LEVEL II ROMs

One of the most fascinating aspects of computers is their versatility. It can do almost anything; all it needs to know is how. With a single ROM change, the TRS-80 can speak in any higher level language we might care to use. In BASIC I, we had just enough mathematical and symbolic capabilities to inspire bigger and better things. The difference between a Level I TRS-80 and a Level II TRS-80 is inspiration and a bigger ROM. Level I used 4K of ROM. In Level II, the system uses 12K of ROM. The added 8K of ROM is enough to give text editing; transcendental functions; give it giant numeric and string arrays; and more system variables than you can use. The guts of the system have not changed; the hardware is the same. The only difference is the machine language contained in ROM.

A Level II machine may be identified by a separate ROM Board stuck to the etch side of the CPU Board. This Board contains three 4K ROMs, a TTL decoder, and a ribbon cable. The cable attaches the majority of the ROM's address inputs and all of the data outputs to the now empty main Board ROM sockets. There is also a 4-conductor ribbon cable (green, orange, red and yellow) coming from the Level II Board. The conductors connect to the CPU Board at A11, A12, A13 and ROM*. The conductors enable a BASIC I to BASIC II conversion on any level production Board to be quite painless.

A11 is used as an address for all three of the ROM's. It is tied to pin 18 of Z1, Z2 and Z3. A12 and A13's leads go to

the A0 and A1 inputs to decoder, Z4. Z4 is an addition to the address decoder network on the main Board. When A12 and A13 is 00, pin 1 of Z4 goes low and ROM A is enabled. When A12 and A13 is 01, ROM B is selected; and when A12 and A13 are 10, ROM C is selected by a low at pin 3. Since we want the ROM's to be accessed only when the CPU needs instruction, ROM* is therefore brought into Z4 to act as a master enable. Only when ROM* is low will we ever select ROM A through ROM C. ROM power, minor ROM addressing and data output is handled by the large ribbon cable. One end of this cable will be placed into the socket of the ROM Board while the other end will be attached in Z31's socket. Addressing and data output are handled by the same circuits that support BASIC I.

**LEVEL I LOGIC BOARD
PARTS LIST**

SYMBOL	DESCRIPTION	PART NUMBER	SYMBOL	DESCRIPTION	PART NUMBER
	PRINTED CIRCUIT BOARD, LOGIC	1700069	C51	0.1 μ F, 10%, 12V, Disc	1500052
			C52	0.1 μ F, 10%, 12V, Disc	1500052
	CAPACITORS		C53	0.1 μ F, 10%, 12V, Disc	1500052
C1	220 μ F, 16V, Electrolytic, Axial	1500059	C54	0.1 μ F, 10%, 12V, Disc	1500052
C2	10 μ F, 16V, Electrolytic, Radial	1500012	C55	0.1 μ F, 10%, 12V, Disc	1500052
C3	0.01 μ F, 10%, 25V, Disc	1500047	C56	0.1 μ F, 10%, 12V, Disc	1500052
C4	10 μ F, 16V, Electrolytic, Radial	1500012	C57	10 μ F, 16V, Electrolytic, Radial	1500012
C5	10 μ F, 16V, Electrolytic, Radial	1500012		DIODES	
C6	100 μ F, 16V, Electrolytic, Radial	1500014	CR1	1N4735, 10%, 6.2V, Zener	4800021
C7	0.01 μ F, 10%, 25V, Disc	1500047	CR2	1N5231, 5%, 5.1V, Zener	4800022
C8	2,200 μ F, 35V, Electrolytic, Axial	1500064	CR3	1N4148, 75V	4800002
C9	10,000 μ F, 16V, Electrolytic, Axial	1500058	CR4	1N4148, 75V	4800002
C10	10 μ F, 16V, Electrolytic, Radial	1500012	CR5	1N4148, 75V	4800002
C11	10 μ F, 16V, Electrolytic, Radial	1500012	CR6	1N4148, 75V	4800002
C12	470 pF, 50V, Disc	1500057	CR7	1N4148, 75V	4800002
C13	470 pF, 50V, Disc	1500057	CR8	Bridge Rectifier, MDA202, 2A, 202V	4800023
C14	0.01 μ F, 10%, 25V, Disc	1500047	CR9	1N982, 75V, Zener	4800026
C15	0.01 μ F, 10%, 25V, Disc	1500047	CR10	1N982, 75V, Zener	4800026
C16	0.1 μ F, 10%, 12V, Disc	1500052		JACKS	
C17	0.1 μ F, 10%, 12V, Disc	1500052	J1	Connector, Socket, DIN, 5 Pin	2100033
C18	0.1 μ F, 10%, 12V, Disc	1500052	J2	Connector, Socket, DIN, 5 Pin	2100033
C19	0.1 μ F, 10%, 12V, Disc	1500052	J3	Connector, Socket, DIN, 5 Pin	2100033
C20	330 pF, 10%, 50V, Disc	1500062		RELAYS	
C21	750 pF, 10%, 50V, Disc	1500050	K1	5V Relay	4500001
C22	0.1 μ F, 10%, 12V, Disc	1500052		TRANSISTORS	
C23	0.1 μ F, 10%, 12V, Disc	1500052	Q1	2N3904, NPN	4822001
C24	220 pF, 10%, 50V, Disc	1500061	Q2	MPS3906, PNP	4822003
C25	220 pF, 10%, 50V, Disc	1500061	Q3	TIP29, Driver	4820004
C26	0.047 μ F, 100V, Polyester Film	1500051	Q4	2N6594, Power	4824003
C27	0.022 μ F, 100V, Polyester Film	1500023	Q5	MPS3906, PNP	4822003
C28	0.1 μ F, 10%, 50V, Disc	1500053	Q6	MJE34, Power	4824002
C29	0.1 μ F, 10%, 12V, Disc	1500052		RESISTORS	
C30	0.1 μ F, 10%, 50V, Disc	1500053	R1	68 ohm, 1/2W, 5%	4708022
C31	0.1 μ F, 10%, 12V, Disc	1500052	R2	2.7 K, 1/4W, 5%	4704056
C32	0.1 μ F, 10%, 50V, Disc	1500053	R3	750 ohm, 1/4W, 5%	4704044
C33	0.1 μ F, 10%, 12V, Disc	1500052	R4	0.33 ohm, 2W, 5%	4717004
C34	0.1 μ F, 10%, 50V, Disc	1500053	R5	1K Trim Pot, 30%	4750019
C35	0.1 μ F, 10%, 12V, Disc	1500052	R6	1.2 K, 1/4W, 5%	4704049
C36	0.1 μ F, 10%, 12V, Disc	1500052	R7	1.2 K, 1/4W, 5%	4704049
C37	0.1 μ F, 10%, 12V, Disc	1500052	R8	100 K, 1/4W, 5%	4704087
C38	0.1 μ F, 10%, 12V, Disc	1500052	R9	3.3 K, 1/4W, 5%	4704058
C39	0.1 μ F, 10%, 12V, Disc	1500052	R10	1 K, Trim Pot, 30%	4750019
C40	0.1 μ F, 10%, 12V, Disc	1500052	R11	3.3 K, 1/4W, 5%	4704058
C41	0.1 μ F, 10%, 12V, Disc	1500052	R12	3.3 K, 1/4W, 5%	4704058
C42	22 μ F, 16V, Electrolytic, Radial	1500055			
C43	47 pF, 10%, 50V, Disc	1500048			
C44	0.1 μ F, 10%, 12V, Disc	1500052			
C45	0.1 μ F, 10%, 12V, Disc	1500052			
C46	0.1 μ F, 10%, 12V, Disc	1500052			
C47	0.1 μ F, 10%, 12V, Disc	1500052			
C48	0.1 μ F, 10%, 12V, Disc	1500052			
C49	0.1 μ F, 10%, 12V, Disc	1500052			
C50	0.1 μ F, 10%, 12V, Disc	1500052			

**LEVEL I LOGIC BOARD
PARTS LIST (Cont'd)**

SYMBOL	DESCRIPTION	PART NUMBER	SYMBOL	DESCRIPTION	PART NUMBER
R13	2.2 K, 1/4W, 5%	4704054			
R14	12 K, 1/4W, 5%	4704070			
R15	1.5 K, 1/4W, 5%	4704050			
R16	1.2 K, 1/4W, 5%	4704049	S1	4PDT Push	5102008
R17	2 K, 1/4W, 5%	4704053	S2	DPDT Push	5102009
R18	5.6 ohm, 3W, 5%	4717003			
R19	220 ohm, 1/2W, 5%	4708032			
R20	100 K, Trim Pot, 20%	4750018			
R21	100 K, Trim Pot, 20%	4750018	Sink Q4	Heatsink	5300003
R22	75 ohm, 1/4W, 5%	4704023	Sink Q6	14 Heatsink	5300002
R23	120 ohm, 1/4W, 5%	4704027			
R24	680 K, 1/4W, 5%	4704100			
R25	1.6 Megohm, 1/4W, 5%	4704106			
R26	1 Megohm, 1/4W, 5%	4704102	X3	16 Pin I.C. Socket	2100037
R27	330 ohm, 1/4W, 5%	4704036	X13	16 Pin I.C. Socket	2100037
R28	270 ohm, 1/4W, 5%	4704034	X14	16 Pin I.C. Socket	2100037
R29	1.8 K, 1/4W, 5%	4704052	X15	16 Pin I.C. Socket	2100037
R30	47 ohm, 1/4W, 5%	4704019	X16	16 Pin I.C. Socket	2100037
R31	10 ohm, 1/4W, 5%	4704011	X17	16 Pin I.C. Socket	2100037
R32	10 K, 1/4W, 5%	4704068	X18	16 Pin I.C. Socket	2100037
R33	360 K, 1/4W, 5%	4704098	X19	16 Pin I.C. Socket	2100037
R34	470 K, 1/4W, 5%	4704097	X20	16 Pin I.C. Socket	2100037
R35	470 K, 1/4W, 5%	4704097	X32	24 Pin I.C. Socket	2100034
R36	360 K, 1/4W, 5%	4704098	X33	24 Pin I.C. Socket	2100034
R37	560 K, 1/4W, 5%	4704099	X39	40 Pin I.C. Socket	2100035
R38	470 K, 1/4W, 5%	4704097	X71	16 Pin I.C. Socket	2100037
R39	4.7 K, 1/4W, 5%	4704061			
R40	4.7 K, 1/4W, 5%	4704061			
R41	470 K, 1/4W, 5%	4704097			
R42	1.0 Megohm, 1/4W, 5%	4704102	Y1	10.6445 MHz, 0.004%, Series Res.	2300004
R43	10 K, 1/4W, 5%	4704068			
R44	10 K, 1/4W, 5%	4704068			
R45	470 K, 1/4W, 5%	4704097			
R46	910 ohm, 1/4W, 5%	4704046	Z1	723, DIP, Voltage Regulator	3100001
R47	10 K, 1/4W, 5%	4704068	Z2	723, DIP, Voltage Regulator	3100001
R48	4.7 K, 1/4W, 5%	4704061	Z4	LM3900, Dual Input Norton Amp.	3100002
R49	4.7 K, 1/4W, 5%	4704061	Z5	74C00 CMOS, Quad 2-Input NAND Gate	3102026
R50	4.7 K, 1/4W, 5%	4704061	Z6	74C04 CMOS, Hex Inverter	3102027
R51	4.7 K, 1/4W, 5%	4704061	Z7	74LS74, Dual D Positive-Edge-Triggered Flip-Flop with Preset and Clear	3102015
R52	910 Ohm, 1/4W, 5%	4704046	Z8	Z4LS153, Dual 4-Line to 1-Line Data Selector/Multiplexer	3102019
R53	1.2 K, 1/4W, 5%	4704049	Z9	74LS04, Hex Inverter	3102008
R54	7.5 K, 1/4W, 5%	4704066	Z10	74LS166, 8-Bit Parallel In/Serial Out Shift Register	3102021
R55	7.5 K, 1/4W, 5%	4704066	Z11	74LS166, 8-Bit Parallel In/Serial Out Shift Register	3102021
R56	220 K, 1/4W, 5%	4704092	Z12	74LS93, Divide by 8 Binary Counter Selector/Multiplexer	3102017
R57	4.7 K, 1/4W, 5%	4704061	Z21	74LS156, Dual 2-Line to 4-Line Decoder/Demultiplexer	3102028
R58	4.7 K, 1/4W, 5%	4704061	Z22	74LS367, TRI-STATE Hex Buffer	3102024
R59	4.7 K, 1/4W, 5%	4704061	Z23	74LS32, Quad 2-Input OR Gate	3102014
R60	4.7 K, 1/4W, 5%	4704061	Z24	74LS132, Quad 2-Input NAND Gate	3102018
R61	4.7 K, 1/4W, 5%	4704061	Z25	74LS32, Quad 2-Input OR Gate	3102014
R62	4.7 K, 1/4W, 5%	4704061			
R63	4.7 K, 1/4W, 5%	4704061			
R64	330 ohm, 1/4W, 5%	4704036			
R65	10 K, 1/4W, 5%	4704068			
R66	4.7 K, 1/4W, 5%	4704061			
*R67	100 ohm, 1/4W, 5%	4704025			

*This Part Number is shown in the Level II (BASIC II) Kit (26-1120) and will be installed in the "G" level Board at time of manufacture.

**KEYBOARD
PARTS LIST**

**LEVEL II KIT
PARTS LIST**

SYMBOL	DESCRIPTION	PART NUMBER	SYMBOL	DESCRIPTION	PART NUMBER
	PRINTED CIRCUIT BOARD, KEYBOARD	1700070		PRINTED CIRCUIT BOARD, LEVEL II ROM ADAPTER	1700081
CAPACITORS			J1	Socket, I.C., 24 Pin	2100034
C1	0.1 μ F, 10%, 12V, Disc	1500052	R1	Resistor, 4.7K, 1/4 W, 5%	4704061
C2	0.1 μ F, 10%, 12V, Disc	1500052	Z1	I.C., 4K x 8 ROM, 450ns, ROM A	3108013
DIODES			Z2	I.C., 4K x 8 ROM, 450ns, ROM B	3108014
CR1	LED, HP5082—4850, Red	2400025	Z3	I.C., 4K x 8 ROM, 450ns, ROM C	3108015
KEYBOARD			Z4	I.C., 74LS42, BCD to Decimal Decoder	3102036
KB1	DS5300, 53 Key, 2—Shot Key caps	5100013			
RESISTORS					
R1	4.7 K, 1/4W, 5%	4704061			
R2	4.7 K, 1/4W, 5%	4704061			
R3	4.7 K, 1/4W, 5%	4704061			
R4	4.7 K, 1/4W, 5%	4704061			
R5	4.7 K, 1/4W, 5%	4704061			
R6	4.7 K, 1/4W, 5%	4704061			
R7	4.7 K, 1/4W, 5%	4704061			
R8	4.7 K, 1/4W, 5%	4704061			
R9	330 ohm, 1/4W, 5%	4704036			
INTEGRATED CIRCUITS					
Z1	74LS05, Hex Buffer with open collector High Voltage outputs	3102009			
Z2	74LS05, Hex Buffer with open collector High Voltage outputs	3102009			
Z3	74LS368, TRI-STATE Hex Buffer	3102025			
Z4	74LS368, TRI-STATE Hex Buffer	3102025			
WIRE					
W1	Stranded, Prebonded, LED, Red, 10"	6002526			
W2	Stranded, Prebonded, LED, Black, 10"	6000526			

BASIC I ROMS

Since the TRS-80 went into production, there were three major PCB changes. These changes reflect different vendor's responses to system requirements as it pertains to the ROM (Read Only Memory). In the manufacturing process, certain vendors were able to supply ROMs to the factory at different times. Hence there are three major types of ROMs. There were slight PCB Modifications as each ROM was used.

INTEL EPROMS

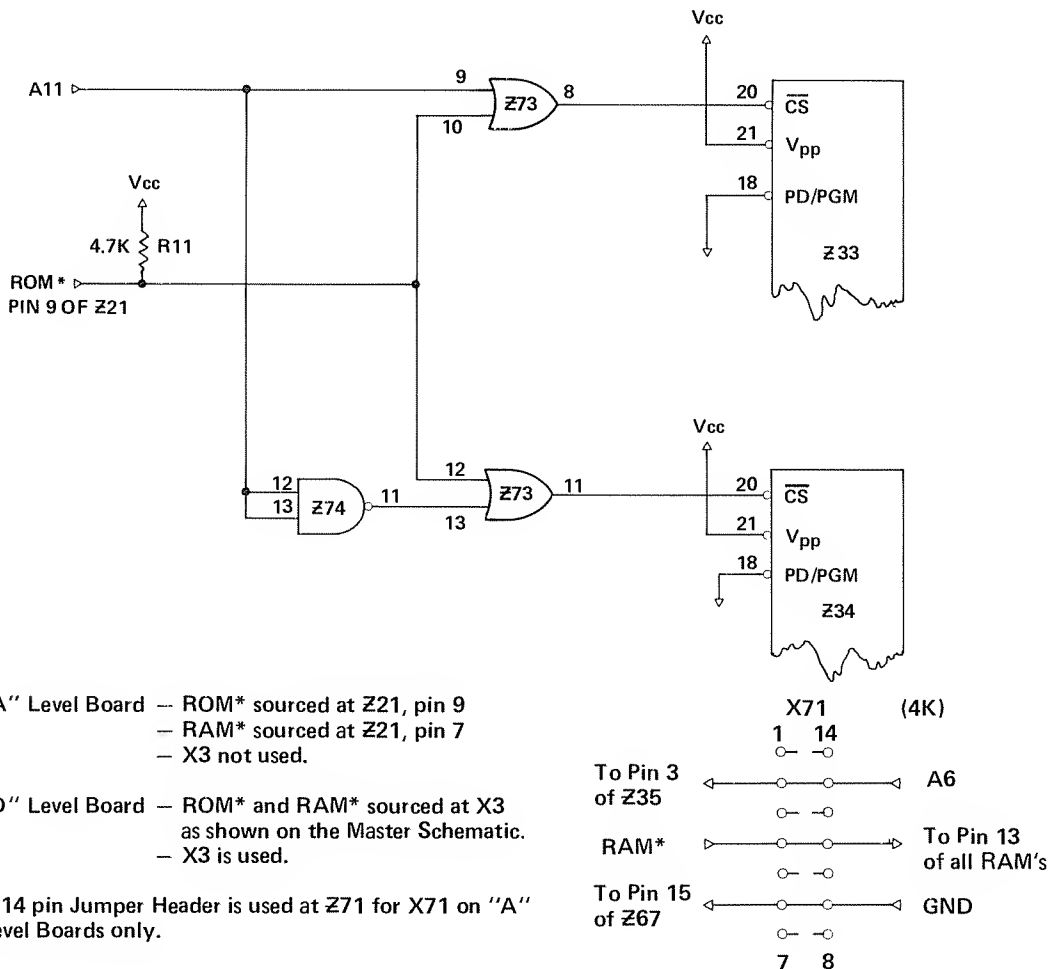
The first mass-produced ROM type was the Intel 2616 EPROM (Erasable, Programmable, Read Only Memory). An Intel ROM may be identified by part number, the Intel trademark and the white painted crystal window. There are two Board levels that use the Intel ROM.

The first Board level may be identified by the "A" right after the part number on the etched side. For example, a Board marked "TRS-80 1700069A" is an "A" level Board. It may also be identified by major wire modifications in two areas. With the PCB upside down and the DIN plugs facing away from you, an "A" Board has several wires on the right side between you and the DIN plugs. It also has wiring to the left of the ROMs, closest to the CPU (Central Processing Unit, the Z-80). The wire mods on the right side are connections in the video, horizontal, and vertical sync

generation areas. The wire mods to the left are changes in the Board to use the Intel EPROMs. Whenever a Board uses the Intel EPROM set, there will always be some added circuitry associated with the \overline{CS} (read "not chip select") pins for each device. This added circuitry uses spare gates on the CPU Board. Figure 14 shows the circuitry differences.

The second Board level that used Intel EPROMs was the "D" level, which may be identified by the letter "D" after the part number. For example, a Board marked "TRS-80 1700069D" is a "D" level Board. It may also be identified by major wire modifications in only one area. These wire mods also utilize the spare gates on the Board for proper \overline{CS} action. The wiring changes for this level Board is the same as for an "A" level Board except some of the wiring needed on the "A" Board is not used on the "D" level because of etch pattern changes.

Since both the "A" and "D" level Boards use the Intel EPROM, it should be noted that each EPROM contains lettering that identifies "ROM A" or "ROM B". ROM A must be in Z33's socket; ROM B must be in Z34's socket. No exceptions! Also, it should be noted that both level Boards have etch cuts. The "A" level Board has many visible etch cuts while the "D" level has only a few. Do not try to "correct" these cuts. The Board will not operate without them.



Note 1: "A" Level Board — ROM* sourced at Z21, pin 9
 — RAM* sourced at Z21, pin 7
 — X3 not used.

"D" Level Board — ROM* and RAM* sourced at X3
 as shown on the Master Schematic.
 — X3 is used.

Note 2: A 14 pin Jumper Header is used at Z71 for X71 on "A" Level Boards only.

Figure 14. Spare Gate Usage on "A" or "D" Level Boards

NATIONAL MM2316 ROMS

The National ROMs were the second type of ROM the factory received during production. These ROMs were used in a large number of "D" level Boards and in a few "A" level Boards. These devices are true Read Only Memories since they are not programmable (except by the vendor), nor are they erasable. They may be identified by the colored ceramic package (only a few) or in 24 pin dual-in-line plastic packages. The ceramic packages have the part number MM2316 followed by either an "R/D" or an "S/D". The "R/D" is ROM A and the "S/D" is ROM B. The plastic packaged ROM has the part number "M2316E" and below it "MMS258ET" followed by an "R/N" or an "S/N". Once again, the "R/N" is ROM A, while the "S/N" is ROM B. These ROMs may be in either Z33 or Z34 socket and still operate correctly. In other words, ROM A does not have to be in socket Z33. It could be in socket Z34, and vice versa for ROM B. The reason you need to know the differences between ROM A and ROM B is in troubleshooting. Certain software troubleshooting aids may fail ROM A and pass ROM B. You need to identify ROM A to replace it or for further troubleshooting.

The "D" level Board with National ROMs will usually have only two wire modifications or jumpers, directly under the ROM sockets. These jumpers will also have two etch cuts associated with them. "A" level Boards using National ROMs may have more jumpers. This is because an "A" level Board may have been modified to use Intel EPROMs and then later remodified to accept the National ROMs. Be careful when identifying the different Board levels. The two wire modifications associated with National ROMs are shown in Figure 15. Notice that only two wires are switched around. There are no spare gates used in this modification.

MOTOROLA 7800 SERIES ROMS (TWO CHIP SET)

Motorola was the next vendor to supply ROMs for the TRS-80. These ROMs may be identified by the part number 7807 for ROM A and 7804 for ROM B. These devices are used only on "D" level Boards and there are no PCB modifications. As with National ROMs, Motorola ROMs may be placed in either Z33's or Z34's socket.

MOTOROLA 7800 SERIES ROMS (SINGLE CHIP SET)

The last ROM supplied to the factory was a single Motorola ROM. It may be packaged in either ceramic (like a few of the National's) or in plastic. The device's number is "7809" and also says "BASIC I". This chip may be inserted in

either Z33 or Z34, but it's usually put in Z33. Once again, there are no PCB modifications. Notice the part number on the single chip ROM and the 7809 ROM B mentioned above. Be very careful when replacing ROMs.

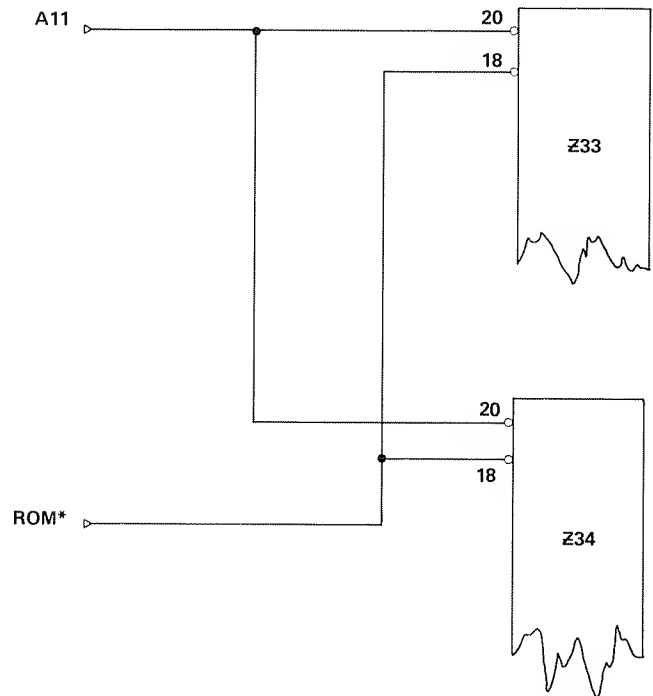


Figure 15. National ROMs in "D" Level Board*

NOTE: The only differences between the Schematic on this page and the Master Schematic are that A11 is on Pin 20 and ROM is on Pin 18.

BASIC II ROMS

The TRS-80 having BASIC II ROMs are easily identified. There are no ROMs plugged into Z33 or Z34. Instead, there is a flat ribbon cable interconnecting the CPU Board to a small 4-chip ROM Board. This ROM Board is stuck, using double-sided tape, to the etched side of the CPU Board. The three ROMs on this Board contain BASIC II. These ROMs may be supplied by various vendors. Figure 16 is the Level II BASIC Schematic.

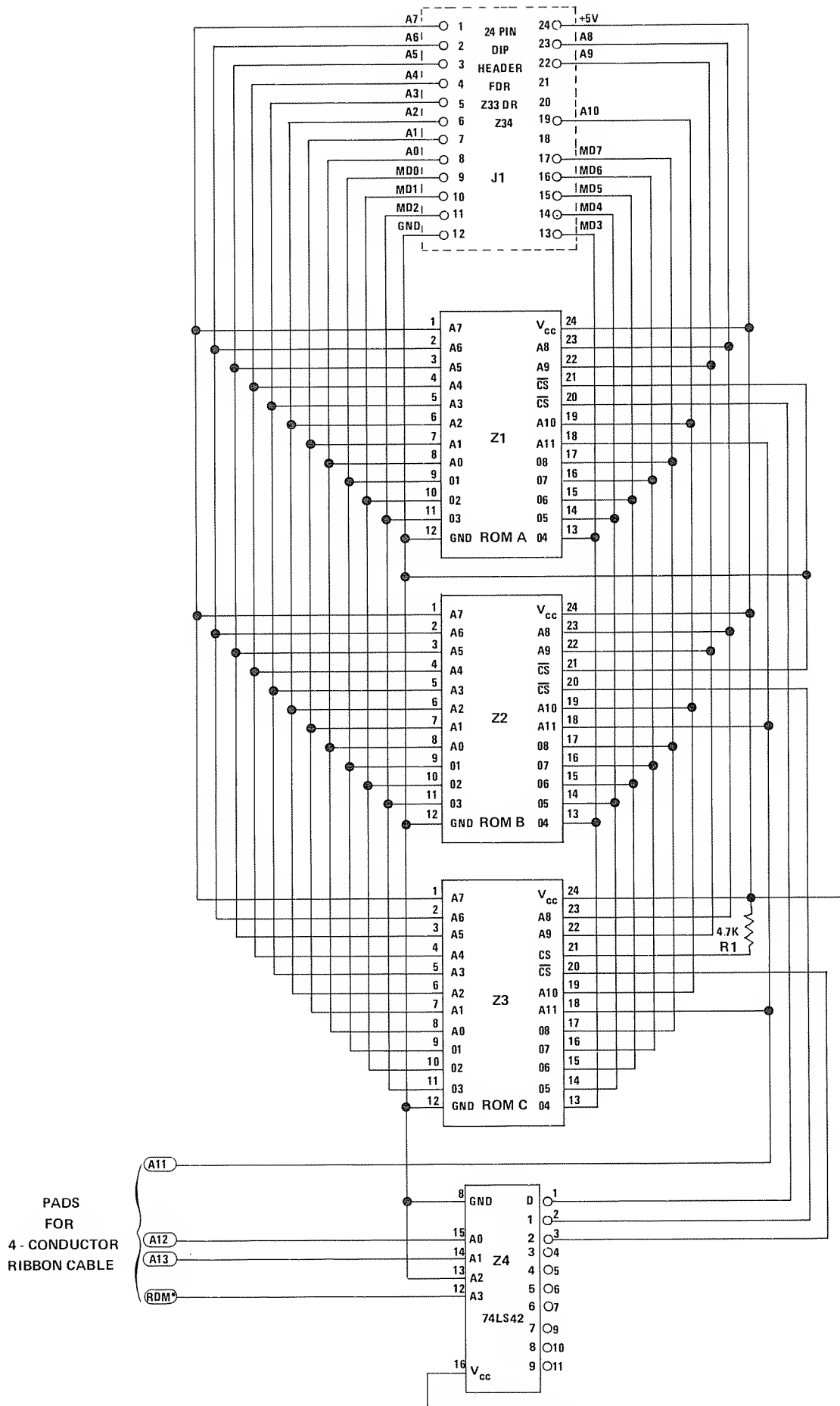


Figure 16. Level II BASIC Schematic

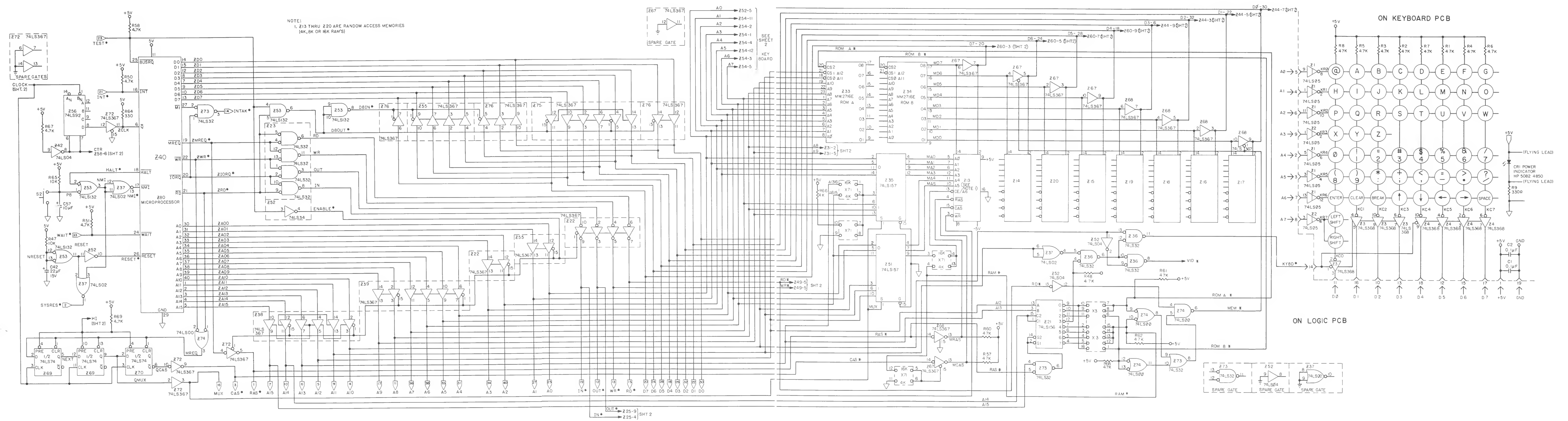



Figure 17. TRS-80 Schematic (Sheet 1)

RADIO SHACK  A DIVISION OF TANDY CORPORATION
U.S.A.: FORT WORTH, TEXAS 76102
CANADA: BARRIE, ONTARIO L4M 4W5

TANDY CORPORATION

AUSTRALIA

280 316 VICTORIA ROAD
RYDALMERE N.S.W. 2116

BELGIUM

PARC INDUSTRIEL DE NANINNE
5140 NANINNE

U.K.

BILSTON ROAD WEDNESBURY
WEST MIDLANDS WS10 7JN